

# Randomness in Complexity Theory and Logics

## DISSERTATION

zur Erlangung des akademischen Grades

DOCTOR RERUM NATURALIUM  
im Fach Informatik

eingereicht an der  
Mathematisch-Naturwissenschaftlichen Fakultät II  
Humboldt-Universität zu Berlin

von  
**Dipl.-Math. Kord Eickmeyer**  
20.08.1979, Lage, Germany

Präsident der Humboldt-Universität zu Berlin:  
Prof. Dr. Jan-Hendrik Olbertz

Dekan der Mathematisch-Naturwissenschaftlichen Fakultät II:  
Prof. Dr. Elmar Kulke

Gutachter:

1. Prof. Dr. Martin Grohe
2. Prof. Dr. Nicole Schweikardt
3. Prof. Dr. Peter Bro Miltersen

**Tag der mündlichen Prüfung:** 29. August 2011

## Abstract

This thesis is comprised of two main parts whose common theme is the question of how powerful randomness as a computational resource is. In the first part (chapter 2) we deal with random structures such as graphs or families of functions and explain how these can possess – with high probability – properties that can be exploited by computer algorithms. Though it may seem counterintuitive at first, it can be very hard to deterministically construct a structure (such as a graph) possessing some desirable property such as good expansion which a random structure has with high probability. We review some cases where such deterministic constructions have indeed been obtained, and add two new results of this kind: We derandomise a randomised reduction due to Alekhovich and Razborov by constructing certain unbalanced bipartite expander graphs, and we give a reduction from a problem concerning bipartite graphs to the problem of computing the minmax-value in three-player games. The latter reduction had been conceived by Hansen and Verbin in a randomised form, the derandomisation is a contribution of this thesis.

In the second part (chapters 3 and 4), we study the expressive power of various logics when they are enriched by random relation symbols. Our goal is to apply techniques from descriptive complexity theory to the study of randomised complexity classes, and indeed we show that our randomised logics do capture complexity classes under study in complexity theory. Using strong results on the expressive power of first-order logic and the computational power of bounded-depth circuits, we give both positive and negative derandomisation results for our logics. On the negative side, we show that randomised first-order logic gains expressive power over standard first-order logic even on structures with a built-in addition relation. Furthermore, it is not contained in monadic second-order logic on ordered structures, nor in infinitary counting logic on arbitrary structures. On the positive side, we show that randomised first-order logic can be derandomised on structures with a unary vocabulary and is contained in monadic second-order logic on additive structures.

The definition of randomised logics, as well as our results concerning their expressive power, are contributions of this thesis.

## Zusammenfassung

Die vorliegende Dissertation besteht aus zwei Teilen, deren gemeinsames Thema in der Frage besteht, wie mächtig Zufall als Berechnungsressource ist. Im ersten Teil (Kapitel 2) beschäftigen wir uns mit zufälligen Strukturen wie Graphen oder Familien von Funktionen und zeigen, dass diese – mit hoher Wahrscheinlichkeit – Eigenschaften haben können, die von Computeralgorithmen genutzt werden können. Obwohl es zunächst kontraintuitiv sein mag kann es sehr schwierig sein, eine Struktur (wie z.B. einen Graph) deterministisch zu erzeugen, die eine bestimmte gewünschte Eigenschaft wie etwa gute Expansion hat, obwohl eine zufällige Struktur diese Eigenschaft mit hoher Wahrscheinlichkeit hat. Wir betrachten zunächst einige Fälle, in denen solche deterministischen Konstruktionen tatsächlich durchgeführt wurden, und fügen dem zwei neue Ergebnisse dieser Art zu: Wir derandomisieren eine randomisierte Reduktion von Alekhnovich und Razborov, indem wir bestimmte unbalancierte bipartite Expandergraphen konstruieren, und wir geben eine Reduktion von einem Problem über bipartite Graphen auf das Problem, den minmax-Wert in Dreipersonenspielen zu berechnen. Letztere Reduktion wurde von Hansen und Verbin in randomisierter Form erdacht; die Derandomisierung ist Beitrag dieser Arbeit.

Im zweiten Teil (Kapitel 3 und 4) untersuchen wir die Ausdrucksstärke verschiedener Logiken, wenn sie durch zufällige Relationssymbole angereichert werden. Unser Ziel ist es, Techniken aus der deskriptiven Komplexitätstheorie auf die Untersuchung randomisierter Komplexitätsklassen anzuwenden, und tatsächlich können wir zeigen, dass unsere randomisierten Logiken randomisierte Komplexitätsklassen einfangen, die in der Komplexitätstheorie untersucht werden. Unter Benutzung starker Ergebnisse über die Logik erster Stufe und die Berechnungsstärke von Schaltkreisen beschränkter Tiefe geben wir sowohl positive als auch negative Derandomisierungsergebnisse für unsere Logiken. Auf der negativen Seite zeigen wir, dass randomisierte erststufige Logik gegenüber normaler erststufiger Logik an Ausdrucksstärke gewinnt, sogar auf Strukturen mit einer eingebauten Additionsrelation. Außerdem ist sie nicht auf geordneten Strukturen in monadischer zweitstufiger Logik enthalten, und auch nicht in infinitärer Zähllogik auf beliebigen Strukturen. Auf der positiven Seite zeigen wir, dass randomisierte erststufige Logik auf Strukturen mit einem unären Vokabular derandomisiert werden kann und auf additiven Strukturen in monadischer Logik zweiter Stufe enthalten ist.

Die Definition der randomisierten Logiken sowie die Ergebnisse bezüglich ihrer Ausdrucksstärke sind Beiträge dieser Arbeit.



# Danksagung

Mein Dank gilt zunächst meinem Betreuer Martin Grohe, durch den ich auch über mein eigentliches Thema hinaus sehr viel schöne Informatik kennengelernt habe und dessen Unterstützung gerade in Durststrecken wesentlich für den Erfolg dieser Promotion war. Ein großer Dank gilt auch allen, die am Lehrstuhl Logik in der Informatik beschäftigt sind oder waren und die für die wundervolle Atmosphäre dort gesorgt haben. Danke an André Hernich und Holger Dell, die Teile der Arbeit probegesehen haben.

Ein besonderer Dank gilt dem Team des Kinderladens „Humbolde“ der Humboldt-Universität – ohne Euch hätte sich die Fertigstellung dieser Arbeit noch deutlich weiter verzögert. Kyo wird Euch vermissen!

Was mich, last but not least, zu Yumiko und Kyo bringt, die mir neben viel Unterstützung auch immer wieder gezeigt haben, dass Arbeit nur das halbe Leben ist. Danke!



# Contents

<b>0</b>	<b>Introduction</b>	<b>1</b>
0.1	Contributions of this thesis . . . . .	2
<b>1</b>	<b>Mathematical Preliminaries</b>	<b>5</b>
1.1	Notation . . . . .	5
1.2	Probability Theory . . . . .	5
1.3	Logics . . . . .	7
1.3.1	Structures and Queries . . . . .	7
1.3.2	Logics . . . . .	9
1.4	Computational Complexity . . . . .	12
1.4.1	Randomisation Operators . . . . .	13
1.4.2	Randomised Polynomial Time . . . . .	13
1.4.3	The Complexity Class $AC^0$ . . . . .	14
1.4.4	Uniform $AC^0$ . . . . .	17
1.4.5	Randomised $AC^0$ . . . . .	18
1.5	Descriptive Complexity Theory . . . . .	19
1.6	First-order Logic and Bounded Depth Circuits . . . . .	21
<b>2</b>	<b>Random and Pseudorandom Structures</b>	<b>23</b>
2.1	The Probabilistic Method . . . . .	23
2.1.1	Colour Coding and Perfect Hash Functions . . . . .	24
2.1.2	Schöning's Algorithm . . . . .	25
2.2	Inapproximability of Weighted Monotone Circuit Satisfiability . . . . .	26
2.2.1	Details of the reduction . . . . .	27
2.2.2	Parameterized Inapproximability . . . . .	31
2.3	Inapproximability of the Minmax Value in Three Player Games . . . . .	35
2.3.1	The Minmax Value in Three Player Games . . . . .	35
2.3.2	Our Results . . . . .	37
2.3.3	Related Work . . . . .	40
2.3.4	The randomised reduction . . . . .	41
2.3.5	Derandomisation . . . . .	45
<b>3</b>	<b>Randomised Logics</b>	<b>49</b>
3.1	Randomised logics . . . . .	49
3.2	Previous Work on Randomised Logics . . . . .	51
3.3	Capturing Results . . . . .	52
3.3.1	BPFO Captures $BPAC^0$ on Ordered Structures . . . . .	53

## Contents

3.3.2	A Logic Capturing BPP . . . . .	54
3.4	Separation Results . . . . .	56
3.4.1	RFO is Not Contained in $C_{\infty\omega}^\omega$ . . . . .	57
3.4.2	BPFO on Ordered Structures is Not Contained in MSO . . . . .	61
3.4.3	RFO is Stronger than FO on Additive Structures . . . . .	64
<b>4</b>	<b>Derandomising Logics</b> . . . . .	<b>67</b>
4.1	BPFO $\equiv$ FO on Unary Vocabularies . . . . .	68
4.2	BPFO is Contained in MSO on Additive Structures . . . . .	73
4.3	Randomised First-Order Logic on Words . . . . .	79



# 0 Introduction

Randomness has been used by algorithm designers already in the early stages of computer science. One very surprising early use of randomness was in Miller and Rabin’s primality test [CLRS01], published in 1976: Given a number  $n$  in binary, to decide whether or not it is a prime number, it guesses a random number  $r \in [0, n - 1]$  and tries to use it to get a certificate for  $n$ ’s non-primality. While no  $r$  will give a certificate of non-primality if  $n$  is in fact a prime number, for every composite number  $n$  at least half the possible values of  $r$  will yield a certificate. Given access to random bits (so it can actually guess  $r$ ), the algorithm runs in time polynomial in the length of  $n$ ’s binary representation.

While modern computers come equipped with hardware devices for generating random bits, the question of in how far randomness as a resource can be substituted by, say, more running time, is nearly as old as the use of randomness itself: The problem of computing sequences of numbers which “look random” has already been studied by von Neumann in the 1940s (according to [Knu81, p. 3]), and Knuth treated this problem thoroughly in the second volume of *The Art of Computer Programming* [Knu81]. In the case of the Miller-Rabin primality test, the question of whether there is a deterministic algorithm for testing primality in polynomial time was open for nearly thirty years, until in 2002 such an algorithm was found by Agrawal et al. [AKS04]. One important problem for which a randomised polynomial time algorithm but no deterministic one is known is polynomial identity testing; see [Sax09] for an overview.

The question of whether derandomisation is possible comes in two essentially different flavours: On the one hand, one may ask for a specific randomised algorithm whether it can be derandomised at, say, a polynomial increase in running time. On the other hand, one may ask for a randomised complexity class such as BPP whether it can be derandomised or not.

Both questions have been extensively studied, and very different tools were developed to tackle them. For derandomising a given algorithm, deterministic constructions of structures have been devised which can be substituted for random ones in many applications. For example, instead of truly random bits,  $k$ -wise independent bits or even almost  $k$ -wise independent bits suffice for many applications, and various constructions of these have been obtained [NN93, AGHP92]. Other important examples of pseudorandom structures include families of perfect hash functions [AYZ95] and expander graphs [GG81, RVW00].

As for derandomising complexity classes, the question of whether BPP is equal to PTIME or not is considered by many to be of equal importance as the question of whether NP is equal to PTIME or not, and progress on it has, so far, culminated in a conditional derandomisation by Impagliazzo and Wigderson [IW97], which says that  $\text{BPP} = \text{PTIME}$  if there is a language in  $\text{DTIME}(2^{O(n)})$  requiring circuits of size  $2^{\Omega(n)}$ .

This is a typical example of a *hardness versus randomness* result, others being Viola’s conditional derandomisation of BPAC<sup>0</sup> [Vio04] and Klivans and van Melkebeek’s results for Arthur-Merlin games [KvM02].

## 0.1 Contributions of this thesis

In chapter 2 we present two new derandomisations of given algorithms: We prove that there is gap-introducing fpt reduction from  $p$ -WSAT(CIRC<sup>+</sup>) to itself and that there is a polynomial-time Turing reduction from Gap-DBS to Gap-minmax.<sup>1</sup>

In the first case, we derandomise a reduction which Alekhovich and Razborov used in [AR01] to show that resolution is not automatisable (i.e., that there is no deterministic algorithm which, given a non-satisfiable formula  $\varphi$  of propositional logic, outputs a resolution proof of this non-satisfiability and running in time polynomial in the length of the shortest such proof). They first showed that resolution is not automatisable unless there is a 2-approximation algorithm for  $p$ -WSAT(CIRC<sup>+</sup>), the problem of determining the minimum weight of a satisfying assignment to a monotone Boolean circuit. They then give a randomised gap-introducing reduction with fpt running time from  $p$ -WSAT(CIRC<sup>+</sup>) to itself to obtain, from a randomised fpt 2-approximation algorithm, a randomised fpt algorithm for solving  $p$ -WSAT(CIRC<sup>+</sup>) exactly. As this is a W[P]-hard problem, they conclude that resolution is not automatisable unless  $W[P] = FPR$  (randomised FPT).

By derandomising their gap-introducing reduction we succeeded in weakening their assumption to  $W[P] = FPT$ , a much more standard assumption. Alekhovich and Razborov’s reduction used certain unbalanced bipartite graphs with good expansion properties. They noticed that random graphs have these properties with high probability but did not give an explicit construction. Upon inspection of their proof, we found that, in fact, weaker expansion properties suffice, and gave an explicit construction of graphs with these weaker properties. Our construction is elementary, and similar constructions had been used before as so-called *Nisan-Wigderson designs*; see, for example, [NW88] and [BMRV00].

The second derandomisation we give in chapter 2 yields a polynomial time Turing reduction from Gap-DBS to Gap-minmax. Because there is evidence towards the fact that Gap-DBS is not decidable in PTIME, the same can be conjectured about Gap-minmax, which would mean that the minmax value in three player games can not be approximated up to an additive error in PTIME. The reduction has been conceived by Hansen and Verbin in a randomised form, the derandomisation is a contribution of this thesis. It uses families of perfect hash functions constructed by Alon et al. [AYZ95].

Chapters 3 and 4 deal with randomised logics, which we introduce in order to apply tools from descriptive complexity theory to the study of randomised complexity classes. These had not been studied previously, though there had been some conceptually similar research which we review in section 3.2. Therefore, all results in these chapters are results of original research, partly together with Martin Grohe.

---

<sup>1</sup>All of these problems will be defined in chapter 2.

In particular, we prove capturing results which show that randomised logics can indeed be used to derive results about certain randomised complexity classes, most notably about  $\text{BPAC}^0$  under various uniformity conditions. While most randomised complexity classes under study in complexity theory are more or less believed to be derandomisable, we prove unconditionally that randomised logics do gain expressive power in some cases. This translates into the new result that  $\text{FO}[+]$ -uniform  $\text{BPAC}^0$  can *not* be derandomised (theorem 39).

On the other hand, we are able to prove non-definability results for randomised logics, in particular for  $\text{BPFO}$  on structures with a unary vocabulary (section 4.1), and we prove containment of  $\text{BPFO}$  in  $\text{MSO}$  on additive structures (section 4.2). Non-definability results for randomised logics correspond to lower bounds for randomised computation, and both are rather hard to obtain. In section 4.3 we show how Ehrenfeucht-Fraïssé games can be used to obtain such results in a very restricted setting.



# 1 Mathematical Preliminaries

## 1.1 Notation

Most of our notation is fairly standard. For a set  $X$ , we denote its powerset by  $2^X$  and its cardinality by  $|X|$ . The symbol  $\mathbb{N}$  denotes the set  $\{0, 1, 2, \dots\}$  of natural numbers including 0, and  $\mathbb{R}$  denotes the set of reals. For natural  $n \geq 1$ , we set

$$[n] := \{1, 2, \dots, n\},$$

and for  $a \leq b \in \mathbb{N}$ , we set

$$[a, b] := \{a, a + 1, \dots, b\}.$$

For a set  $\Sigma$ , we denote by  $\Sigma^k$  the set of  $k$ -tuples  $(\sigma_1, \dots, \sigma_k)$  of elements of this set. We also call these tuples *words*, in which case we write them without commas or brackets as  $\sigma_1 \dots \sigma_k$ ; the set  $\Sigma$  will be called *alphabet* in this context and will usually be a finite set. The empty word will be denoted by  $\lambda$ , thus  $\Sigma^0 = \{\lambda\}$ . The set

$$\Sigma^* := \bigcup_{k=0}^{\infty} \Sigma^k$$

is the set of all words of arbitrary length. Though we may speak of tuples and words interchangeably, speaking of words emphasises the existence of a binary operation on  $\Sigma^*$ , namely concatenation (written by juxtaposition  $uv$  for words  $u, v \in \Sigma^*$ ); this operation is associative and therefore turns  $\Sigma^*$  into a monoid with  $\lambda$  as neutral element. Exponentiation is to be understood as repeated concatenation, so

$$u^k := \underbrace{uu \cdots u}_{k \text{ times}}$$

for  $u \in \Sigma^*$  and  $k \in \mathbb{N}$ . The *length* of a word  $w$  is written by  $|w|$ , so  $|w| = k$  for  $w \in \Sigma^k$ .

We denote by  $\log x$  the logarithm with base 2, and by  $\ln$  the natural logarithm.

## 1.2 Probability Theory

We need some fundamental notions and results from probability theory. Since we only need discrete probability spaces, no measure-theoretic foundations are necessary. An excellent treatment of this part of probability theory can be found in Feller's classic text [Fel57].

## 1 Mathematical Preliminaries

A (discrete) *probability space* consists of an at most countable set  $\Omega$  and a function  $\mathbb{P} : 2^\Omega \rightarrow [0, 1]$  such that

- $\mathbb{P}(\emptyset) = 0, \mathbb{P}(\Omega) = 1,$
- $\mathbb{P}(\bigcup_i A_i) = \sum_i \mathbb{P}(A_i)$  for disjoint  $A_1, A_2, \dots \subseteq \Omega$

Subsets  $A \subseteq \Omega$  are called *events*, the quantity  $\mathbb{P}(A)$  is called the *probability* of  $A$ . For  $\omega \in \Omega$  we write  $p_\omega$  for  $\mathbb{P}(\{\omega\})$ . In particular,

$$\mathbb{P}(A) = \sum_{\omega \in A} p_\omega$$

for every  $A \subseteq \Omega$ . For (not necessarily disjoint) events  $A_1, A_2, \dots$  the *union bound* applies:

$$\mathbb{P}\left(\bigcup_{i \geq 1} A_i\right) \leq \sum_{i \geq 1} \mathbb{P}(A_i).$$

A function  $X : \Omega \rightarrow M$  is called ( $M$ -valued) *random variable*. Every such random variable defines a probability measure on  $\text{im}(X) = \{X(\omega) \mid \omega \in \Omega\}$  by

$$\mathbb{P}(A) := \mathbb{P}(X^{-1}(A)),$$

and this is called the *distribution* of the random variable  $X$ . For real-valued random variables, we define the *expected value* to be

$$\mathbb{E}X := \sum_{\omega \in \Omega} p_\omega X(\omega).$$

By  $\{X < \alpha\}$  we denote the set

$$\{\omega \in \Omega \mid X(\omega) < \alpha\}$$

with  $\alpha \in \mathbb{R}$ , and similar for  $\{X \leq \alpha\}$  and so on. As is customary, we drop the set brackets in expressions like  $\mathbb{P}(X < \alpha)$ .

Two events  $A, B \subseteq \Omega$  are called *independent* if

$$\mathbb{P}(A \cap B) = \mathbb{P}(A) \cdot \mathbb{P}(B),$$

and two random variables  $X, Y : \Omega \rightarrow \mathbb{R}$  are called independent if the events  $\{X > x\}$  and  $\{Y > y\}$  are independent for all  $x, y \in \mathbb{R}$ . For a finite set  $\Omega$ , the *uniform distribution* is the probability measure defined by  $\mathbb{P}(\{\omega\}) = |\Omega|^{-1}$  for all  $\omega \in \Omega$ . More generally, for every  $p \in [0, 1]$  we can define a probability measure on  $\Omega = 2^{[n]}$  by

$$p_A := p^{|A|}(1-p)^{n-|A|}$$

for every  $A \subseteq [n]$ . In other words, the random variable  $S : \Omega \rightarrow 2^{[n]}, A \mapsto A$  on this probability space satisfies  $\mathbb{P}(i \in S) = p$  for all  $i \in [n]$ , and the events  $\{i \in S\}$  and

$\{j \in S\}$  are independent for all  $i \neq j \in [n]$ . Thus,  $S$  is a random subset of  $[n]$  such that each  $i \in [n]$  is in  $S$  independently with probability  $p$ . The size  $X = |S|$  of this random subset is again a random variable, whose distribution satisfies

$$\mathbb{P}(X = k) = \binom{n}{k} p^k (1-p)^{n-k}.$$

This distribution is called the  $(n, p)$ -binomial distribution, and its expected value is  $\mathbb{E}X = pn$ . We will frequently invoke the following theorem due to Chernoff:

**Theorem 1** (Chernoff's Tail Bound). *Let  $X$  be distributed according to the  $(n, p)$ -binomial distribution. Then*

$$\mathbb{P}(X > (1 + \delta)np) < \left( \frac{e^\delta}{(1 + \delta)^{(1 + \delta)}} \right)^{np}$$

for all  $\delta > 0$ . Furthermore, for  $0 < \delta \leq 1$ ,

$$\mathbb{P}(X < (1 - \delta)np) < \exp\left(-\frac{np\delta^2}{2}\right)$$

and

$$\mathbb{P}(X > (1 + \delta)np) < \exp\left(-\frac{np\delta^2}{4}\right).$$

For a proof of this theorem, see [MR95, chapter 4].

## 1.3 Logics

We will mostly be concerned with first-order predicate logics and some extensions thereof. We review basic concepts of these logics here; details can be found, e.g., in [EFT96] or [EF99].

### 1.3.1 Structures and Queries

In this thesis we will only be concerned with relational structures, thus a *vocabulary*  $\sigma$  is a finite set containing relation symbols, which we commonly denote by uppercase letters  $R, S$  and so on. Each relation symbol has an associated arity  $r \geq 1$ . A  $\sigma$ -structure consists of a non-empty set  $V(A)$  and sets  $R(A) \subseteq V(A)^r$  for each symbol  $R \in \sigma$  of arity  $r$ . The set  $V(A)$  is also called the *universe* of the structure  $A$ . We use the symbol  $\cong$  to denote isomorphism of structures, i.e.,  $A \cong B$  iff there is a bijection  $f : V(A) \rightarrow V(B)$  such that

$$(a_1, \dots, a_r) \in R(A) \quad \text{iff} \quad (f(a_1), \dots, f(a_r)) \in R(B)$$

for all  $r$ -ary relation symbols  $R$  and all  $a_1, \dots, a_r \in V(A)$ .

## 1 Mathematical Preliminaries

Let  $\sigma, \tau$  be vocabularies with  $\sigma \subseteq \tau$ . Then the  $\sigma$ -restriction of a  $\tau$ -structure  $B$  is the  $\sigma$ -structure  $B|_\sigma$  with universe  $V(B|_\sigma) := V(B)$  and relations  $R(B|_\sigma) := R(B)$  for all  $R \in \sigma$ . A  $\tau$ -expansion of a  $\sigma$ -structure  $A$  is a  $\tau$ -structure  $B$  such that  $B|_\sigma = A$ . For every class  $\mathcal{C}$  of structures,  $\mathcal{C}[\tau]$  denotes the class of all  $\tau$ -structures in  $\mathcal{C}$ . A renaming of a vocabulary  $\tau$  is a bijective mapping  $r$  from  $\tau$  to a vocabulary  $\tau'$  such that for all  $R \in \tau$  the relation symbol  $r(R) \in \tau'$  has the same arity as  $R$ . If  $r : \tau \rightarrow \tau'$  is a renaming and  $A$  is a  $\tau$ -structure then  $A^r$  is the  $\tau'$ -structure with  $V(A^r) := V(A)$  and  $r(R)(A^r) := R(A)$  for all  $R \in \tau$ .

We let  $+1, \leq, +, \times$  and  $\text{Bit}$  be distinguished relation symbols of arity two, two, three, three, and two, respectively. Whenever any of these relations symbols appear in a vocabulary  $\tau$ , we demand that they be interpreted by a successor relation, linear order, ternary addition and multiplication relations, and bit-relation respectively, in all  $\tau$ -structures. To be precise, we denote by  $\mathcal{N}_n$  the  $\{+1, \leq, +, \times, \text{Bit}\}$ -structure with

$$\begin{aligned} V(\mathcal{N}_n) &= [0, n-1], & +1(\mathcal{N}_n) &= \{(a, a+1) \mid 0 \leq a \leq n-2\}, \\ \leq(\mathcal{N}_n) &= \{(a, b) \mid a \leq b\}, & +(\mathcal{N}_n) &= \{(a, b, c) \mid a+b=c\}, \text{ and} \\ \times(\mathcal{N}_n) &= \{(a, b, c) \mid a \cdot b = c\} & \text{Bit}(\mathcal{N}_n) &= \{(a, b) \mid [a]_b = 1, \text{ where } a = \sum [a]_i 2^i \\ & & & \text{is the binary representation of } a\} \end{aligned}$$

We demand  $A|_{\{+1, \leq, +, \times, \text{Bit}\} \cap \tau} \cong (\mathcal{N}_{|A|})|_{\{+1, \leq, +, \times, \text{Bit}\} \cap \tau}$  for all  $\tau$ -structures  $A$ . We call structures whose vocabulary contains any of these relation symbols *with successor relation, ordered, additive, multiplicative, and with bit predicate*, respectively. By  $\mathcal{O}_n$  we denote a linear order with  $n$  elements, i.e.,  $\mathcal{O}_n$  is the  $\{\leq\}$ -structure  $\mathcal{N}_n|_{\{\leq\}}$ .

An important class of structures is the class of *word structures*. Given a finite alphabet  $\Sigma$  and a nonempty subset  $\tau \subseteq \{+1, \leq, +, \times, \text{Bit}\}$ , let

$$\tau_\Sigma := \tau \cup \{P_s \mid s \in \Sigma\},$$

where the  $P_s$  are distinct unary relation symbols. A  $\tau_\Sigma$ -structure  $W$  is a word structure if every  $x \in V(W)$  is in exactly one of the  $P_s(W)$ . Notice that because we assume that the relations in  $\tau$  are interpreted the same way as in some  $\mathcal{N}_n$ , if at least one of  $+, \leq, +1$  and  $\text{Bit}$  are in  $\tau$ , then these relations induce a linear ordering on  $V(W)$ , though if  $\leq \notin \tau$ , our logics may not necessarily be able to speak about this ordering. However, because of this ordering there is a (up to isomorphisms) one-to-one correspondence between word structures and strings in  $\Sigma^*$ . In this case, we call  $\tau$  a *valid set of arithmetic relations*, and denote by  $w_x^{(\tau)}$  the word structure corresponding to  $x \in \Sigma^*$  (for definiteness, we take the one with universe  $\{1, \dots, |x|\}$  and arithmetic relations as in  $\mathcal{N}_n$ ).

A  $k$ -ary  $\tau$ -global relation is a mapping  $\mathcal{R}$  that associates a  $k$ -ary relation  $\mathcal{R}(A)$  with each  $\tau$ -structure  $A$ . A 0-ary  $\tau$ -global relation is usually called a *Boolean*  $\tau$ -global relation. We identify the two 0-ary relations  $\emptyset$  and  $\{()\}$ , where  $()$  denotes the empty tuple, with the truth values **false** and **true**, respectively, and we identify the Boolean  $\tau$ -global relation  $\mathcal{R}$  with the class of all  $\tau$ -structures  $A$  with  $\mathcal{R}(A) = \text{true}$ . A  $k$ -ary  $\tau$ -query is a  $k$ -ary  $\tau$ -global relation  $\mathcal{Q}$  preserved under isomorphism, that is, if  $f$  is an isomorphism from



a  $\tau$ -structure  $A$  to a  $\tau$ -structure  $B$  then for all  $\vec{a} \in V(A)^k$  it holds that  $\vec{a} \in \mathcal{Q}(A) \iff f(\vec{a}) \in \mathcal{Q}(B)$ .

### 1.3.2 Logics

A logic  $L$  consists of two parts:

- A *syntax* which assigns to each vocabulary  $\tau$  a set  $L[\tau]$  of  $L$ -formulas of vocabulary  $\tau$ , and
- a *semantics*, which assigns to each formula  $\varphi \in L[\tau]$  a  $\tau$ -global relation  $\mathcal{Q}_\varphi^{L[\tau]}$ . If this relation is  $k$ -ary, we will write

$$A \models \varphi[a_1, \dots, a_k]$$

for  $(a_1, \dots, a_k) \in \mathcal{Q}_\varphi^{L[\tau]}(A)$  for a structure  $A$  and  $a_1, \dots, a_k \in V(A)$ . In case  $\varphi$  is a *sentence*, i.e., a formula for which  $\mathcal{Q}_\varphi^{L[\tau]}$  is a 0-ary relation, we write  $A \models \varphi$  for  $() \in \mathcal{Q}_\varphi^{L[\tau]}$ .

The query  $\mathcal{Q}_\varphi^{L[\tau]}$  is called the query *defined* by  $\varphi$ , and a  $\tau$ -query  $\mathcal{Q}$  is called *definable* in a logic  $L$  if  $\mathcal{Q} = \mathcal{Q}_\varphi^{L[\tau]}$  for some  $\varphi \in L[\tau]$ . Two formulas  $\varphi, \psi \in L[\tau]$  are said to be *equivalent* if they define the same query, written as  $\varphi \equiv \psi$ .

The semantics is supposed to satisfy the following assumptions which are generally accepted as minimal requirements any logic should satisfy (cf. [Ebb85]):

- (i) For all  $\varphi \in L[\tau]$  the global relation  $\mathcal{Q}_\varphi^{L[\tau]}$  is a  $\tau$ -query.
- (ii) If  $\sigma \subseteq \tau$  then  $L[\sigma] \subseteq L[\tau]$ , and for all formulas  $\varphi \in L[\sigma]$  and all  $\tau$ -structures  $A$  it holds that  $\mathcal{Q}_\varphi^{L[\sigma]}(A|_\sigma) = \mathcal{Q}_\varphi^{L[\tau]}(A)$ .
- (iii) If  $r : \tau \rightarrow \tau'$  is a renaming, then for every formula  $\varphi \in L[\tau]$  there is a formula  $\varphi^r \in L[\tau']$  such that for all  $\tau$ -structures  $A$  it holds that  $\mathcal{Q}_\varphi^{L[\tau]}(A) = \mathcal{Q}_{\varphi^r}^{L[\tau']}(A^r)$ .

Though for most of the logics we will be dealing with, the  $\models$ -relation can easily be defined for structures of arbitrary size, we will only be concerned with finite structures. Thus if we say that a certain query consisting of finite structures can not be defined, we mean that there is no formula in that particular logic that holds in a finite structure iff this structure is in the query. This is sometimes called *finite axiomatisability*.

It is customary to require the set  $L[\tau]$  to be recursive (i.e., decidable by some Turing machine) for all  $\tau$ . In this case, we speak of a logic *with decidable syntax*. We will also encounter logics with undecidable syntax in chapter 3.

We say that a formula  $\varphi(x)$  defining a unary query *defines an element* if in every structure it is satisfied by exactly one element. Since we may identify the elements of an ordered structure uniquely with natural numbers it makes sense to say, e.g., that “ $\varphi(x)$  defines a prime number” or “ $\varphi(x)$  defines a number  $\leq \log^{O(1)} |A|$ ”, and we will sometimes do so.

We compare the expressive power of two logics  $L$  and  $L'$  by saying that  $L$  is *weaker* than  $L'$  (written  $L \preceq L'$ ) if every query  $Q$  that can be defined in  $L$  can also be defined in  $L'$ ; we will also use the expression “ $L$  *embeds* into  $L'$ ” for this. We say that  $L$  is *strictly weaker* than  $L'$ , written  $L \prec L'$ , if  $L \preceq L'$  but not  $L' \preceq L$ . We write  $L \equiv L'$  if exactly the same queries are definable in  $L$  and  $L'$ , i.e., if both  $L \preceq L'$  and  $L' \preceq L$ .

We state some logics which will be of importance in the following. All definitions are more or less standard and can be found, e.g., in [EFT96].

**First-Order Logic** Formulas in first-order logic (FO) can be atomic formulas ( $x \doteq y$  for variables  $x$  and  $y$ , relational formulas  $Rx_1 \dots x_r$  for  $r$ -ary relation symbols  $R$  and variables  $x_1, \dots, x_r$ ), Boolean combinations of formulas (using  $\wedge$ ,  $\vee$  and  $\neg$ ), and quantified formulas of the form  $\exists x \varphi$  and  $\forall x \varphi$ .

Strong tools have been developed in finite model theory to prove non-definability in first-order logic. In particular Ehrenfeucht-Fraïssé games can be used to show that certain queries such as connectedness in graphs are not definable in FO. In a broader perspective, theorems like those of Gaifman and Hanf show that first-order logic can only speak about “local” properties of structures, in some precisely definable sense. Details can be found in chapter 2 of [EF99].

**Infinitary Logics** Formulas in the infinitary logic  $L_{\infty\omega}^\omega$  are built up from atomic formulas using Boolean combinations and quantification just like first-order formulas. In addition, we allow infinite conjunctions and disjunctions, i.e., formulas of the form

$$\bigwedge_{i \in I} \varphi_i \quad \text{and} \quad \bigvee_{i \in I} \varphi_i,$$

where  $I$  is an arbitrary index set and the  $\varphi_i$  are  $L_{\infty\omega}^\omega$  formulas themselves. Though the formulas are allowed to be of arbitrary size, we only allow a finite (but arbitrary) number of variables to appear in each formula (otherwise every query of finite structures instantly becomes definable). Interest in  $L_{\infty\omega}^\omega$  (and its counting counterpart  $C_{\infty\omega}^\omega$ , see below) mainly stems from that fact that, while many interesting logics such as fixed-point logics are weaker than  $L_{\infty\omega}^\omega$ , it still has quite severe and provable limitations (such as 0-1-laws and its inability to count) which a fortiori also pertain to any logic weaker than it.

Ehrenfeucht-Fraïssé games can be adapted to the infinitary finite-variable case (i.e.,  $L_{\infty\omega}^\omega$ ) by using so-called *pebble games*. This way, strong non-definability results have been obtained for this logic as well, an example being the query containing all structures with an even-sized universe over the empty vocabulary.

**Counting Logics** First-order logic as well as the infinitary logic  $L_{\infty\omega}^\omega$  lack the ability to count, as witnessed by the fact that, e.g., the query consisting of all sets of even cardinality is not definable in these logics. Several extensions of first-order and infinitary logic have been introduced, a good reference is [Ott96].

Adding counting abilities to logics is complicated by the fact that, while one usually wants these logics to be able to speak about arithmetic relations on numbers, one

does not want to impose a linear order on the input structure. One way of coping with this is by using a two-sorted logic. Thus, variables may either hold a universe element or a number, and a structure with  $n$  elements is enriched with  $n + 1$  elements of the number sort representing the numbers from 0 to  $n$ . We denote universe variables by roman letters  $x, y$ , etc., and number variables by greek letters  $\xi, \zeta$  etc. One usually allows certain arithmetical relations such as  $\leq$ ,  $+$  and  $\times$  on the number sort, which are interpreted with their usual meaning.

Counting may then be introduced via counting terms or counting quantifiers. In the first case, for every formula  $\varphi$  and every universe variable  $x$ , the term

$$\#x \varphi$$

is of the number sort and specifies the number of universe elements  $a$  such that  $\varphi$  is satisfied if  $x$  is interpreted by  $a$ . Note that this may depend on other free variables (both universe and number) in  $\varphi$ . Counting quantifiers, on the other hand, are formulas of the form

$$\exists^{=\xi} x \varphi,$$

which states that exactly  $\xi$  choices for  $x$  will satisfy  $\varphi$ .

Counting variants of infinitary logics are somewhat easier to define, by just adding quantifiers of the form

$$\exists^{\geq n} x \varphi,$$

with *constant*  $n \in \mathbb{N}$ . This is because, in infinitary logics, any relation on the number sort may be spelled out explicitly by a possibly infinite formula.

In section 3.3.2, we will need a very limited form of counting based on so called *Rescher quantifiers*.

**Restricted Variable Logics** We sometimes restrict the number of variables that our formulas may contain. E.g., by  $\text{FO}^k$  we denote the set of all first-order formulas containing only  $k$  distinct variables. By rebinding variables, the quantifier depth may still be arbitrary. The exponent  $\omega$  in the infinitary logics  $\text{L}_{\infty\omega}^\omega$  and  $\text{C}_{\infty\omega}^\omega$  is meant to suggest that formulas in these logics may only use an arbitrarily large but finite number of distinct variables. Restricting the number of variables to a fixed finite number  $k$  is denoted by  $\text{L}_{\infty\omega}^k$  and  $\text{C}_{\infty\omega}^k$ .

**Second-Order Logic** Formulas in second order logic may additionally contain atomic formulas  $Xx_1 \dots, x_r$  for an  $r$ -ary relation variable  $X$  and second-order quantifications  $\exists X \varphi$  and  $\forall X \varphi$ . In *monadic second-order logic* MSO, all relation variables must be unary.

**Fixed-point logics** Fixed-point logics have been introduced in an attempt to define a logic capturing PTIME. We deal only with inflationary fixed-point logic IFP here, which extends first-order logic by fixed-point operators of the following form:

$$[\text{IFP}_{X, \vec{x}} \varphi(X, \vec{x})](\vec{t}),$$

## 1 Mathematical Preliminaries

where  $\varphi$  is itself an IFP formula,  $X$  is a second-order variable,  $\vec{x}$  is a tuple of first-order variables and  $\vec{t}$  a tuple of terms, all of the same arity, say  $r$ . Its semantics is defined as follows: In a structure  $A$ , for each relation  $B \subseteq V(A)^r$ , the formula  $\varphi$  defines a new relation

$$F_\varphi(B) := \{\vec{a} \in V(A)^r \mid A \models \varphi(B, \vec{a})\}.$$

Note that  $F_\varphi(B)$  depends on  $A$  as well as on the interpretation of all free variables in  $\varphi$  other than  $X$  and  $\vec{x}$ . We define a sequence  $(B_k)_{k \geq 1}$  of relations by

$$\begin{aligned} B_0 &= \emptyset, \\ B_{k+1} &= B_k \cup F_\varphi(B_k). \end{aligned}$$

Because  $B_k \subseteq B_{k+1}$ , in finite structures  $A$ , this sequence must become stationary after a finite number of steps, say

$$B_m = B_{m+1} = \dots$$

Then

$$A \models [\text{IFP}_{X, \vec{x}} \varphi(X, \vec{x})](\vec{t})$$

iff the interpretation of  $\vec{t}$  is in  $B_m$ . Note that IFP embeds into  $\mathsf{L}_{\infty\omega}^\omega$ , so any non-definability result for  $\mathsf{L}_{\infty\omega}^\omega$  also holds for IFP.

The various extensions to first-order logic may also be combined. In particular, IFP+C denotes fixed-point logic with counting, i.e., two-sorted logic with either counting quantifiers or counting terms and fixed-point operators. This logic embeds into  $\mathsf{C}_{\infty\omega}^\omega$ , so non-definability results for  $\mathsf{C}_{\infty\omega}^\omega$  also hold for IFP+C.

### 1.4 Computational Complexity

We use standard concepts from computational complexity theory, details may be found, e.g., in [AB09, Weg05, Pap93]. In particular, a *language* is a set  $L \subseteq \Sigma^*$  of words over some finite alphabet  $\Sigma$ , which we will usually assume to be  $\{0, 1\}$ . A *complexity class* is a set of languages, usually defined by giving resource bounds on computing models for deciding them. For a class  $\mathcal{F}$  of functions  $f : \mathbb{N} \rightarrow \mathbb{N}$ , the class  $\text{DTIME}(\mathcal{F})$  is the class of all languages decidable by deterministic Turing machines with running time bounded by one of the functions in  $\mathcal{F}$ , and similarly with  $\text{NPTIME}(\mathcal{F})$  for non-deterministic Turing machines. In particular,

$$\text{PTIME} = \text{DTIME}(n^{O(1)}) \quad \text{and} \quad \text{NP} = \text{NPTIME}(n^{O(1)}).$$

### 1.4.1 Randomisation Operators

We define three operators  $R$ ,  $ZP$  and  $BP$  which work on complexity classes and generate randomised classes from non-randomised ones. For a function  $f : \mathbb{N} \rightarrow \mathbb{N}$  and a language  $L \subseteq \Sigma^*$ , we define a function  $p_{L,f} : \Sigma^* \rightarrow [0, 1]$  by

$$p_{L,f}(x) := \frac{|\{y \in \Sigma^{f(|x|)} \mid \langle x, y \rangle \in L\}|}{|\Sigma|^{f(|x|)}},$$

where  $\langle \cdot, \cdot \rangle : \Sigma^* \times \Sigma^* \rightarrow \Sigma$  is a pairing function. We use this to define

**One-sided bounded error** For  $\alpha \in (0, 1)$ , a language  $L$  is in  $R_\alpha \mathcal{C}$  if there is a language  $M \in \mathcal{C}$  and a polynomially bounded function  $f$  such that

$$p_{M,f}(x) \begin{cases} = 0 & \text{if } x \notin L \\ \geq \alpha & \text{if } x \in L \end{cases}$$

for all  $x \in \Sigma^*$ . It is in  $\text{co-}R_\alpha \mathcal{C}$  if there is a language  $M \in \mathcal{C}$  and a polynomially bounded function  $f$  such that

$$p_{M,f}(x) \begin{cases} \leq \alpha & \text{if } x \notin L \\ = 1 & \text{if } x \in L \end{cases}$$

for all  $x \in \Sigma^*$ . In general, we allow  $\alpha$  to depend on  $|x|$ . If  $\alpha = 1/2$ , we just write  $RC$  and  $\text{co-}RC$ . Thus  $R$  is an operator operating on complexity classes.

**Zero error** We define  $ZPC := RC \cap \text{co-}RC$ .

**Two-sided bounded error** For  $0 < \alpha < \beta < 1$  (which may depend on  $|x|$ ), a language  $L$  is in  $BP_{\alpha,\beta} \mathcal{C}$  if there is a language  $M \in \mathcal{C}$  and a polynomially bounded function  $f$  such that

$$p_{M,f}(x) \begin{cases} \leq \alpha & \text{if } x \notin L \\ > \beta & \text{if } x \in L \end{cases}$$

for all  $x \in \Sigma^*$ . The choice of  $>$  over  $\geq$  for the second condition does not affect the definition for natural choices of  $\mathcal{C}$  and fits in well with our definition of randomised logics in chapter 3. For  $\alpha = 1/3$  and  $\beta = 2/3$  we just write  $BPC$  instead of  $BP_{1/3,2/3} \mathcal{C}$ . Again,  $BP$  is an operator on complexity classes.

The choice of  $1/2$ ,  $1/3$  and  $2/3$  in the above definitions is arbitrary, for most natural complexity classes  $\mathcal{C}$  (e.g., polynomial time), the resulting classes are rather robust with respect to changes in  $\alpha$  and  $\beta$ .

### 1.4.2 Randomised Polynomial Time

The class  $BPP := BP_{1/3,2/3} \text{PTIME}$  of randomised polynomial time with two-sided error has been studied extensively. We first note that  $BPP$  is very robust with respect to

## 1 Mathematical Preliminaries

changing the error bounds of  $1/3$  and  $2/3$  used in the definition of the BP operator. In fact,

$$\text{BP}_{2^{-|x|^c}, 1-2^{-|x|^c}} \text{PTIME} = \text{BPP} = \text{BP}_{\alpha^{-|x|^{-c}}, \alpha+|x|^{-c}} \text{PTIME}$$

for all  $\alpha \in (0, 1)$  and all  $c > 0$ . This is because in polynomial time, we can make polynomially many trials and take a majority vote, see, e.g., [AB09].

By trying all possible values for the random bits, BPP is readily seen to be included in the class

$$\text{EXPTIME} = \text{DTIME}(2^{n^{O(1)}}).$$

Furthermore, by a result of Adleman, BPP is in  $\text{PTIME}_{/\text{poly}}$ , the class of all problem decidable by circuit families of polynomial size; equivalently, this is the class of problems decidable in deterministic polynomial time by machines which get, in addition to their input, an *advice string* of polynomial length which may only depend on the length of the input. By a famous result of Karp and Lipton [KL80], if  $\text{NP} \subseteq \text{PTIME}_{/\text{poly}}$  then the polynomial hierarchy collapses to its second level, thus it is considered unlikely that BPP contains any NP-complete problems.

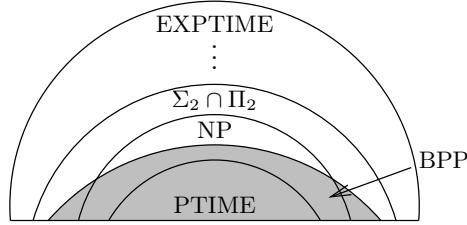


Figure 1.1: Unconditionally known results on BPP.

While it is not known whether  $\text{BPP} \subseteq \text{NP}$ , Sipser and Gács [Sip83] proved that  $\text{BPP} \subseteq \Sigma_2^P \cap \Pi_2^P$ , i.e., BPP is contained in the second level of the polynomial hierarchy. A simpler proof of this fact was given by Lautemann [Lau83] in the same year, and we will rely on ideas from that proof in section 4.2.

Arguably the most important open question concerning BPP is whether it is equal to PTIME or not. Implagliazzo and Wigderson [IW97], building on a long line of work, proved that  $\text{BPP} = \text{PTIME}$  if there is a language decidable in deterministic time  $2^{O(n)}$  which requires circuits of size at least  $2^{\Omega(n)}$ . The proof works by constructing for every  $c$  a pseudorandom generator which, given  $O(\log n)$  truly random bits, computes a string of  $n^c$  pseudorandom bits that looks random to any circuit of size at most  $n^c$ . This result is generally seen as evidence towards the fact that  $\text{BPP} = \text{PTIME}$ . On the other hand, it is conditional under a very strong circuit lower bound, something far beyond current techniques.

### 1.4.3 The Complexity Class $\text{AC}^0$

Instead of giving, as in the case of Turing machines, a single algorithm for inputs of arbitrary sizes, we may also specify a family  $(C_n)_{n \geq 1}$  of Boolean circuits such that each

circuit  $C_n$  has  $n$  inputs and one output. Here, by a (*Boolean*) *circuit* we mean a directed acyclic graph in which each node of in-degree  $> 1$  is labelled as and-node or as or-node, each node of in-degree 1 is labelled as negation node and all nodes of in-degree 0 are input nodes. Furthermore one node with out-degree 0 is labelled as output node. The *size of a circuit*  $C$  is the total number of nodes and edges and is denoted by  $|C|$ . Given an assignment  $a \in \{0, 1\}^n$  for a circuit  $C$  with  $n$  input nodes, we say that  $a$  *satisfies*  $C$  if the value computed by  $C$  on input  $a$  is 1. The *depth* of a circuit is the length of a longest path from its output to one of its input nodes. The *fan-in* of a circuit is the maximal in-degree among its nodes.

**Definition 2.** Given a circuit family  $(C_n)_{n \geq 1}$ , the language accepted by it is the set

$$\{x \in \{0, 1\}^* \mid C_{|x|} \text{ accepts } x\}.$$

The class  $AC^0$  is defined as the class of all languages  $L \subset \{0, 1\}^*$  for which there exists a circuit family  $(C_n)_{n \geq 1}$  which accepts it and such that there is a  $d > 1$  such that all  $C_n$  have depth at most  $d$ , and for which  $|C_n| = n^{O(1)}$ . Note that we do not assume any bound on the fan-in of the  $C_n$ .

Although lower bounds on computational resources have been the core goal of research in computational complexity for several decades now, unconditional results are still very few. The class  $AC^0$  is a notable exception, because Håstad's Switching Lemma for bounded depth-circuits [Hå86] can be used to obtain exponential lower bounds for constant depth circuits. We say that a function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  can be expressed as a  $k$ -DNF if it can be written as

$$f(\mathbf{x}) := \bigvee_i \lambda_{i,1} \wedge \dots \wedge \lambda_{i,k}$$

for some  $i \geq 1$  and some choice of literals  $\lambda_{i,j}$ , each of which is either some  $x_k$  or its negate  $\neg x_k$ . Expressibility as a  $k$ -CNF is defined similarly. Both for CNFs and for DNFs, tight lower bounds are easy to obtain by means of so called *prime-implicants*.

A random  $p$ -restriction  $\rho$  is a tuple  $(\rho_1, \dots, \rho_n)$  of independent random variables such that

$$\mathbb{P}(\rho_i = *) = p \quad \text{and} \quad \mathbb{P}(\rho_i = 0) = \mathbb{P}(\rho_i = 1) = \frac{1-p}{2}.$$

For any outcome  $\rho$  of this random variable, the restricted function  $f|_\rho$  is a function on those variables  $x_i$  for which  $\rho_i = *$ , such that

$$f|_\rho(\mathbf{x}) = f(\mathbf{y}), \text{ where } y_i = \begin{cases} x_i & \text{if } \rho_i = *, \\ \rho_i & \text{otherwise.} \end{cases}$$

The use of random restrictions to obtain lower bounds for bounded-depth circuits has been pioneered by Furst, Saxe, and Sipser, who in [FSS81] used it to prove that the parity function has no  $AC^0$  circuits; the journal version appeared in [FSS84]. Building on the technique of random restrictions, subsequently Ajtai [Ajt83] and Yao [Yao85] obtained

## 1 Mathematical Preliminaries

size lower bounds of  $\Omega(n^{c_d \log n})$  and  $\Omega(2^{n^{1/4d}})$ , respectively, for circuits of depth  $d$ . (Here,  $c_d$  is a constant depending on  $d$ ). In [Hå86], Johan Håstad obtained a lower bound of  $\Omega(2^{c_d n^{1/(d-1)}})$  which is optimal in the sense that for some  $\tilde{c}_d$ , there are depth- $d$  circuit families of size  $O(2^{\tilde{c}_d n^{1/(d-1)}})$  computing the parity function.

The common scheme in proving these lower bounds is the use of so called *switching lemmas* which state that, after applying a random restriction, with high probability each of the subcircuits in the two lowest levels of the circuit (those closest to the input gates) may be switched from  $\wedge$ - $\vee$ -circuits to  $\vee$ - $\wedge$ -circuits of similar size or vice versa. After merging two subsequent layers of gates of the same type, one obtains a circuit of depth one lower than that of the original circuit, until eventually one arrives at a CNF or DNF, for which known lower bounds apply. The switching lemma as proved by Håstad reads:

**Theorem 3** (Håstad's Switching Lemma). *Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be expressible as a  $k$ -CNF, and let  $\rho$  be a random  $p$ -restriction for some  $p \in (0, 1)$ . Then for all  $s$ , the probability that  $f|_\rho$  is expressible by an  $s$ -DNF is at least  $1 - \alpha^s$ , where  $\alpha$  is the unique positive root of the equation*

$$\left(1 + \frac{4p}{(1+p)\alpha}\right)^k = \left(1 + \frac{2p}{(1+p)\alpha}\right)^k + 1$$

An overview of the switching lemma and its applications is given in [Bea94]. The most well-known consequences are the above-mentioned lower bound for the parity function and other functions with known lower bounds on DNFs or CNFs, in particular the majority function.

Another important consequence of Håstad's Switching Lemma is the low average sensitivity of  $AC^0$  circuits:

**Definition 4.** Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be a Boolean function. The *sensitivity* of  $f$  at  $\vec{x} \in \{0, 1\}^n$  is defined as

$$S(f; \vec{x}) := |\{1 \leq i \leq n \mid f(\vec{x}) \neq f(\vec{x} \oplus \vec{e}_i)\}|,$$

where  $\vec{x} \oplus \vec{e}_i$  is the vector  $\vec{x}$  with the  $i$ -th bit flipped. The *average sensitivity* of  $f$  is

$$S(f) := 2^{-n} \sum_{\vec{x}} S(f; \vec{x}).$$

The average sensitivity of  $f$  may be interpreted as follows: Arrange the  $2^n$  elements of  $\{0, 1\}^n$  into a hypercube, connecting two vertices  $\vec{x}, \vec{y} \in \{0, 1\}^n$  by an edge iff they differ in exactly one coordinate. Colour the vertices of this hypercube red and black according to  $f(\vec{x})$ , and call an edge *coloured* if it connects two vertices of different colour. Then  $S(f)/n$  is the probability that a randomly chosen edge in this hypercube is coloured. In the extreme case that  $f$  is the parity function or its complement, all edges are coloured. In this case  $S(f) = S(f; \vec{x}) = n$  for all  $\vec{x} \in \{0, 1\}^n$ .

In [LMN89] Linial, Mansour, and Nisan gave a bound on the average sensitivity of functions computable by  $AC^0$  circuit families, which was later strengthened by Bop-



pana [Bop97]:

**Theorem 5.** *Let  $f : \{0, 1\}^* \rightarrow \{0, 1\}$  be a Boolean function computable by a family  $(C_n)_{n \geq 0}$  of Boolean circuits of depth  $d$  and size  $n^{O(1)}$  consisting of negation gates and  $\vee$  and  $\wedge$ -gates of unbounded fan-in. Then*

$$S(f_n) \leq O(\log^{d-1} n).$$

Note that Boppana's bound is optimal, as the parity of  $\log^{d-1} n$  many input bits may be computed by polynomial-size circuits of depth  $d$ .

Linial et al. used their bound on the average sensitivity of  $AC^0$  functions to give an  $O(n^{\log^{O(1)} n})$ -time algorithm for learning functions in  $AC^0$  [LMN89]. Another important application was found by Rossman [Ros08], who used Boppana's result to prove that bounded-depth circuits for detecting cliques of size  $k$  in graphs must have size at least  $|V|^{2k/9}$ , independent of the bound on their depth. Amano [Ama10] extended this result to arbitrary subgraphs. Another application to finite model theory is given in [Ros09], where Rossman used Boppana's result to show that certain strategies in Ehrenfeucht-Fraïssé games on random structures are winning strategies with high probability.

#### 1.4.4 Uniform $AC^0$

The class  $AC^0$  as defined in Definition 2 is non-uniform in the sense that, while each circuit  $C_n$  in a circuit family can be described by a some finite string, there is no finite string describing the whole circuit family. As a consequence,  $AC^0$  contains even certain non-recursive languages, such as the language

$$\{1^n \mid n \text{ is the Gödel number of a Turing machine which halts on the empty input}\}.$$

On the other hand, the very strong lower bounds resulting from Håstad's Switching Lemma also hold in this non-uniform setting, making the class  $AC^0$  an interesting class nonetheless.

In the context of randomised computation, however, non-uniformity allows for very strong derandomisation results, also in the case of randomised  $AC^0$  (cf. section 1.4.5). We therefore introduce uniform variants of  $AC^0$ , by demanding that each circuit  $C_n$  of a uniform circuit family must be constructible, given  $n$ , by a Turing machine with certain resource bounds. The following definitions are essentially taken from Barrington et al. [BIS90].

Let  $(C_n)_{n \geq 1}$  be a circuit family, and assume in each circuit  $C_n$  the gates are labelled with natural numbers from  $\{0, \dots, m_n\}$  for some  $m_n$ . We assume that the gate with number 0 is the unique output gate in each circuit, and that the gates with numbers  $1, \dots, n$  are the input gates. We define the *direct connection language*  $L_{(C_n)} \subseteq \{0, 1\}^*$  of the family to be the set of all tuples  $\langle t, u, v, y \rangle$  where  $u$  and  $v$  are numbers of gates in  $C_n$  such that  $u$  is of type  $t$  (with  $t = 1$  for  $\vee$ -gates,  $t = 2$  for  $\wedge$ -gates and  $t = 3$  for  $\neg$ -gates), the output of gate  $v$  is an input to gate  $u$ , and  $y = 1^n$ . We assume the numbers  $t$ ,  $u$  and  $v$  to be encoded in binary, and assume a suitable pairing function

$\langle \cdot, \cdot, \cdot, \cdot \rangle : (\{0, 1\}^*)^4 \rightarrow \{0, 1\}^*$ .

For a complexity class  $\mathcal{C}$  and a circuit family  $(C_n)_{n \geq 1}$ , we say that  $(C_n)$  is  $\mathcal{C}$ -uniform if its direct connection language  $L_{(C_n)}$  is in  $\mathcal{C}$ . Accordingly,  $\mathcal{C}$ -uniform  $\text{AC}^0$  is the class of all languages  $L$  which are decidable by a  $\mathcal{C}$ -uniform circuit family of bounded depth and polynomial size. Among the possible choices for  $\mathcal{C}$ , the class dlogtime-uniform  $\text{AC}^0$  stands out as it has a very neat equivalent definition in terms of descriptive complexity theory, cf. section 1.6 and [BIS90]. Because of this, when speaking of uniform  $\text{AC}^0$  without explicitly stating the complexity class  $\mathcal{C}$ , one usually means dlogtime-uniform  $\text{AC}^0$ .

### 1.4.5 Randomised $\text{AC}^0$

Using the BP and R operators defined in section 1.4.1, we can define randomised analogues of  $\text{AC}^0$  in a natural way. By an application of Håstad's Switching Lemma, the majority function is not computable in  $\text{AC}^0$ , i.e., there is no  $\text{AC}^0$  circuit family  $(C_n)_{n \geq 1}$  such that  $C_n$  accepts a string  $x_1 \dots x_n$  iff at least half of the  $x_i$  are 1. Thus, to improve the error probability for randomised  $\text{AC}^0$ -circuits, we cannot just take polynomially many independent copies of each circuit with independent random bits and take the majority of their results. However, for every  $c > 0$  and  $\eta = \Omega(1/\log^c n)$  the following promise-problem is decidable in  $\text{AC}^0$ :

$\eta$ -APPROXIMATE-MAJORITY

*Input:*  $x_1 \dots x_n \in \{0, 1\}^n$   
*Promise:*  $\frac{1}{n} \sum x_i \notin (\frac{1}{2} - \eta, \frac{1}{2} + \eta)$   
*Problem:* Decide whether  $\sum x_i > n/2$ .

For a proof, cf. [Vio11]. Using this result, one can show that

$$\text{BP}_{(2^{-|x|^c}, 1-2^{-|x|^c})} \text{AC}^0 = \text{BPAC}^0 = \text{BP}_{(1/2-\log^{-c}|x|, 1/2+\log^{-c}|x|)} \text{AC}^0$$

for all  $c > 0$ . Because Adleman's result that  $\text{BPP} \subseteq \text{PTIME}_{\text{poly}}$  only depends on the ability to improve the error probability to below  $2^{-|x|}$ , and because  $\text{AC}^0$  is itself non-uniform, these classes are in fact equal to  $\text{AC}^0$ .

Nisan [Nis91] gave a construction for a pseudorandom generator that reduces the number of random bits for a  $\text{BPAC}^0$ -circuit family to polylogarithmically many. We will use this construction in section 4.2.

Applying the BP operator to  $\mathcal{C}$ -uniform of  $\text{AC}^0$  results in the class  $\mathcal{C}$ -uniform  $\text{BPAC}^0$ . While  $\text{PTIME}$ -uniform  $\text{AC}^0$  is easily seen to be in  $\text{PTIME}$ , it is still not known whether  $\text{PTIME}$ -uniform  $\text{BPAC}^0$  is also in  $\text{PTIME}$ . By using Nisan's pseudorandom generator and trying all possible random seeds,  $\text{PTIME}$ -uniform  $\text{BPAC}^0$  is seen to be in the class  $\text{DTIME}(n^{\log^{O(1)} n})$ .

The question of whether dlogtime-uniform  $\text{BPAC}^0$  can be derandomised is still open, but there is a conditional derandomisation similar to Impagliazzo and Wigderson's result for  $\text{BPP}$  by Viola [Vio04]. Here, the condition is that there is a language  $L$  which is

decidable in alternating linear time with a constant number of alternations but which is hard on average for circuits of linear size, in the sense that every circuit of size  $2\ell$  fails to compute  $L \cap \{0, 1\}^\ell$  on at least an inverse polynomial fraction of inputs.

## 1.5 Descriptive Complexity Theory

Descriptive complexity theory relates the expressive power of logics to the computational power of complexity classes. Good introductory texts are [Imm99, Lib04, EF99].

In order to define what it means for a Boolean query  $\mathcal{Q}$  of structures to be decidable in some complexity class  $\mathcal{C}$ , we encode a structure  $A$  over an arbitrary vocabulary  $\sigma$  into a string of length polynomial in  $|V(A)|$  by stating, for each relation symbol  $R \in \sigma$  and each tuple  $\vec{a}$  of elements in  $A$ , whether  $\vec{a} \in R(A)$  or not. We may assume some ordering on the vocabulary  $\sigma$ , say  $\sigma = \{R_1, \dots, R_k\}$  with arities  $r_1, \dots, r_k$ , and that the string  $w_A$  encoding the structure  $A$  is just a juxtaposition  $1^{|A|} 0 w_A^{(1)} w_A^{(2)} \dots w_A^{(k)}$ , where

$$w_A^{(i)} \in \{0, 1\}^{|V(A)|^{r_i}}$$

encodes relation  $R_i$ , with one position for each tuple. The prefix  $1^{|A|} 0$  is added to avoid degeneracy in the case of the empty vocabulary. This still leaves the question of how each individual relation should be encoded, i.e., the ordering of the tuples. Using lexicographic ordering, it suffices to fix an ordering  $\leq$  on the universe  $V(A)$ ; call the resulting string  $w_A^\leq$ . This way we get a set of strings

$$S(A) := \{w_A^\leq \mid \leq \text{ is a linear order on } V(A)\},$$

and we say the query  $\mathcal{Q}$  is decidable in the complexity class  $\mathcal{C}$  iff the language

$$L_{\mathcal{Q}} := \bigcup_{A \in \mathcal{Q}} S(A)$$

is in  $\mathcal{C}$ . This leads us to the following definition:

**Definition 6.** Let  $\mathcal{C}$  be a complexity class, i.e., a set of decision problems  $D \subset \{0, 1\}^*$ , and  $\mathbb{L}$  a logic. We say that  $\mathbb{L}$  *captures*  $\mathcal{C}$  if

(C1) For every Boolean query  $\mathcal{Q}$  decidable in  $\mathcal{C}$  there is a sentence  $\varphi \in \mathbb{L}$  such that

$$A \in \mathcal{Q} \quad \Leftrightarrow \quad A \models \varphi$$

(C2) For every sentence  $\varphi \in \mathbb{L}$ , the query  $\text{Mod}(\varphi)$  is decidable in  $\mathcal{C}$ , and there is a computable function which, given a sentence  $\varphi \in \mathbb{L}$  outputs a Turing machine  $M_\varphi$  satisfying the resource bounds for  $\mathcal{C}$  which decides  $\text{Mod}(\varphi)$ .

This definition corresponds to conditions (C1) and (C2') in Grohe's survey [Gro08]. There is a weaker notion of (C2) which does not require the mapping  $\varphi \mapsto M_\varphi$  to be

## 1 Mathematical Preliminaries

computable, while in the case of resource-bounded complexity classes  $\mathcal{C}$  it is reasonable to strengthen condition (C2) to also require a computable resource bound for  $M_\varphi$ .

The relevance of this definition is witnessed by a number of capturing results, the most famous of which is due to Fagin [Fag74]:

**Theorem 7.** *The logic  $\Sigma_1$  (i.e., existential second-order logic) captures the complexity class NP.*

The question of whether or not there is a logic with decidable syntax that captures PTIME is still open, cf. [Gro08]. There are, however, capturing results for PTIME on restricted classes of structures, in the sense of the following definition:

**Definition 8.** Let  $\mathcal{C}$  be a complexity class,  $\mathbf{L}$  a logic and  $\mathcal{S}$  a Boolean query. We say that  $\mathbf{L}$  *captures  $\mathcal{C}$  on  $\mathcal{S}$*  if

(C1) For every Boolean query  $\mathcal{Q}$  decidable in  $\mathcal{C}$  there is a sentence  $\varphi \in \mathbf{L}$  such that

$$A \in \mathcal{Q} \iff A \models \varphi$$

for all  $A \in \mathcal{S}$ .

(C2) There is a computable function which maps every sentence  $\varphi \in \mathbf{L}$  to a Turing machine  $M_\varphi$  such that for all  $A \in \mathcal{S}$  and  $x \in S(A)$ ,

$$M_\varphi \text{ accepts } x \iff A \models \varphi.$$

The most prominent example of a capturing result for PTIME on a restricted class of structures is the following result, which was found independently by Immerman [Imm82] and Vardi [Var82]:

**Theorem 9.** *Inflationary fixed-point logic IFP captures PTIME on the class of all ordered structures.*

Recently, Laubner [Lau10] proved that IFP+C captures PTIME on the class of interval graphs.

When the class  $\mathcal{S}$  is the class of all word-models (with a given valid set of arithmetic relations) over a vocabulary  $\Sigma$ , there is a natural correspondence between  $\mathcal{S}$  and  $\Sigma^*$ . In this case, we may restrict ourselves to encoding word-models by the unique string which they encode, and get the following definition:

**Definition 10.** Let  $\mathcal{C}$  be a complexity class,  $\mathbf{L}$  a logic and  $\tau \subseteq \{+1, \leq, +, \times, \text{Bit}\}$  be a valid set of arithmetic relations. We say that  $\mathbf{L}$  *captures  $\mathcal{C}$  on  $\tau$ -word models* if

(C1) For every  $D \in \mathcal{C}$  there is a sentence  $\varphi \in \mathbf{L}$  such that

$$x \in D \iff w_x^{(\tau)} \models \varphi$$

for all  $x \in \Sigma^*$ .

- (C2) There is a computable mapping that takes every  $\varphi \in \mathcal{L}$  to a Turing machine  $M_\varphi$  such that

$$M_\varphi \text{ accepts } x \quad \text{iff} \quad w_x^{(\tau)} \models \varphi$$

for all  $x \in \Sigma^*$ .

## 1.6 First-order Logic and Bounded Depth Circuits

The expressive power of first-order logic on arbitrary structures is rather limited. In fact, on structures over the empty vocabulary or over vocabularies with only unary predicates, FO can only count up to some constant. In particular, the class of all even structures is not definable by any FO sentence.

To enhance the expressive power of first-order logic, two approaches seem natural. One is to extend the logic itself, by introducing new quantifiers such as the counting quantifiers of section 1.3. Another one is to restrict attention to structures with certain pre-defined relations, which may only depend on the size of the structures. Using the second approach, Barrington et al. obtained the following capturing result:

**Theorem 11** (Barrington-Immerman-Straubing [BIS90]). *On the class of all structures with addition and multiplication, first-order logic captures dlogtime-uniform  $AC^0$ .*

In other words, for every dlogtime-uniform circuit family  $(C_n)_{n \geq 1}$  of bounded depth and polynomial size, there is an FO-sentence  $\varphi$  such that for all  $x \in \{0, 1\}^*$ ,

$$C_{|x|} \text{ accepts } x \quad \text{iff} \quad w_x^{(+, \times)} \models \varphi,$$

and for every FO-sentence  $\varphi$  there is a dlogtime-uniform circuit family  $(C_n)_{n \geq 1}$  such that for every ordered structure  $A$ ,

$$C_{|w_A|} \text{ accepts } w_A \quad \text{iff} \quad A \models \varphi,$$

where  $w_A$  is the canonical encoding of  $A$  into a string using the given ordering on  $A$ . Because the bit predicate Bit can be defined in first-order logic using addition and multiplication and vice versa, we may as well state theorem 11 using the bit predicate instead of addition and multiplication. It is because of this neat capturing result that dlogtime-uniformity is generally considered the “right” notion of uniformity for  $AC^0$  circuit families.

Barrington et al. stated their result in more generality by considering also extensions of first-order logic with various quantifiers. Another extension to Theorem 11 is given by Behle and Lange:

**Theorem 12** (Behle and Lange [BL06]). *On the class of ordered structures, FO captures  $FO[\leq]$ -uniform  $AC^0$ , and on the class of additive structures, FO captures  $FO[+]$ -uniform  $AC^0$ .*



## 2 Random and Pseudorandom Structures

In this chapter we will describe how certain properties of random structures can be exploited algorithmically, and show how in some cases structures with these properties can be constructed explicitly.

Perhaps the most fundamental result showing that random structures can have a high degree of order to them is the law of large numbers, which can be phrased as follows: For every  $\epsilon > 0$ , a random string  $x$  drawn uniformly at random from among all strings in  $\{0, 1\}^n$  will have, with high probability, between  $n(1/2 - \epsilon)$  and  $n(1/2 + \epsilon)$  many 1s. Here, “with high probability” means that the probability tends to 1 as  $n$  goes to infinity, though the speed of this convergence depends on the choice of  $\epsilon$ ; very strong bounds for the speed of this convergence are given by Chernoff’s Theorem (Thm. 1). If we call a string with between  $n(1/2 - \epsilon)$  and  $n(1/2 + \epsilon)$  many 1s “nearly balanced”, then this theorem shows that one way to construct a nearly balanced string is to just draw one at random, if one has access to randomness.

While this may not sound too exciting, given that such a string can easily be constructed by, say, a LOGSPACE-bounded Turing machine on input  $n$  in unary, even this very basic result is of some use, e.g., if  $n$  independent agents were to construct a nearly balanced string among themselves. In light of the law of large numbers, if each such agent just flips a coin to decide its letter of the string, with high probability the agents will collectively determine a good string.

In other cases, random structures enjoy desirable properties that are not as easy to obtain deterministically as in the above example. We will give several examples of this in section 2.1 and show how deterministic constructions have been obtained. We will then proceed to show two so-called *gap introducing reductions* which are used to prove non-approximability under certain hardness conditions and which originally relied on properties of certain random structures. As a consequence, the hardness assumptions in these cases involve randomised complexity classes, namely, that certain problems are not solvable by randomised algorithms with some additional resource bounds.

In both cases, we introduce deterministic constructions of objects with the properties in question, and in this way derandomise the reductions. As a consequence, we obtain the same non-approximability results under possibly weaker assumptions concerning only deterministic classes.

### 2.1 The Probabilistic Method

Suppose we have a collection  $\mathcal{C}$  of discrete objects, and a property  $P$  which may hold of an object or not. For example,  $\mathcal{C}$  could be the set of all undirected graphs on  $n$  labelled

vertices, and  $P$  could be the property of having good expansion. Objects satisfying  $P$  will be called *good* objects. Often we would like to show that a good object exists and, if so, construct it.

In many cases, the easiest – or even the only – known way of showing that a good object exists is to show that a randomly chosen object has a strictly positive probability of being good. This technique is commonly called the “probabilistic method” and was pioneered by Erdős, who first used it to prove the existence of a graph with both high chromatic number and high girth [Erd59]. This approach has by now been highly refined; a good introductory text is [AS92].

Often, the probabilistic method gives stronger bounds on the probability of a random object being good than just proving that it is non-zero. If this probability is, say, at least  $\frac{1}{2}$ , then a randomised algorithm which can sample (approximately) uniformly from  $\mathcal{C}$  can construct an object which is good with probability at least  $\frac{1}{2}$ . If, in addition, the algorithm can within its resource bounds check whether an object is good or not, it can repeatedly draw objects until it finds a good one. The expected number of tries until success is constant.

Still, we would like to get an explicit construction of a good object, not just a mere proof of existence. The notion of “explicit construction” is, of course, non-rigorous and depends on the context in which it is used. In theoretical computer science, we usually seek a deterministic algorithm which constructs the desired object within certain resource bounds such as logarithmic space or polynomial time. Note that in particular, we are satisfied with a brute-force search for a “good” object, as long as it can be carried out within the given resource bounds.

We sketch some applications of this method in algorithms and their derandomisations:

### 2.1.1 Colour Coding and Perfect Hash Functions

In [AYZ95], Alon et al. gave algorithms to decide the existence of simple paths and cycles of a given length  $k$  in a graph. These algorithms assign colours from the set  $[k]$  to the vertices uniformly at random, and then use standard algorithmic techniques to decide the existence of a *colourful* path or cycle of length  $k$ , i.e., one in which each of the  $k$  vertices has a different colour. Such a path must necessarily be simple.

Let  $f : [n] \rightarrow [k]$  be a random function drawn uniformly from among all such functions. For any set  $S \subseteq [n]$  of size  $k$ , the probability that  $f$  is injective on  $S$  is given by

$$\begin{aligned} \mathbb{P}(f|_S \text{ injective}) &= 1 \left(1 - \frac{1}{k}\right) \cdots \left(1 - \frac{k-1}{k}\right) \\ &= k^{-k} \cdot k! \\ &\geq e^{-k} \end{aligned}$$

by the inequality  $e^k \geq k^k/k!$  for all  $k \in \mathbb{N}$  (this is just the  $k$ -th term of the power series for  $e^x$ ). The probability that out of  $\ell$  independently drawn functions  $f_1, \dots, f_\ell$  not one



is injective is therefore at most

$$(1 - e^{-k})^\ell \leq \exp(-\ell e^{-k}),$$

which is at most  $1/2$  if  $\ell > (\ln 2)e^k$ . This way, Alon et al. obtain a randomised algorithm running in time  $2^{O(k)} \cdot n^{O(1)}$  which always answers “no” if no simple  $k$ -path exists and which answers “yes” with probability at least  $1/2$  if such a path exists.

This algorithm has been derandomised in the same paper by constructing *families of perfect hash functions*. A family  $\mathcal{F}$  of functions  $f: [n] \rightarrow [k]$  is called a family of perfect hash functions if for every  $S \subseteq [n]$  of size  $k$  at least one of the  $f \in \mathcal{F}$  is injective on  $S$ . Suppose  $\mathcal{F}$  consists of  $\ell$  independently and uniformly drawn functions. As we saw above, for each *fixed*  $S$ , the probability that some  $f \in \mathcal{F}$  is injective on  $S$  is at least  $\exp(-\ell e^{-k})$ . There are  $\binom{n}{k}$  many such  $S$ , and by the union bound, the probability that for every  $S$  one of the functions  $f \in \mathcal{F}$  is injective is at least

$$\binom{n}{k} \exp(-\ell e^{-k}) \leq \exp(k \ln n - \ell e^{-k}),$$

and this is greater than zero if  $\ell > k \cdot e^k \cdot \ln n$ , which proves that families of perfect hash functions of this size exist. Alon et al. also gave an explicit (i.e., computable in time polynomial in the size of the family) construction of such a family of size  $2^{O(k)} \cdot \ln n$ , which we will need in section 2.3.

In 2007, Alon and Gutner gave a construction of so called *balanced families of perfect hash functions* which even allow approximate counting of simple paths; cf. [AG07]. Furthermore, the ideas used in colour coding can be applied used to detect more general substructures than simple paths or cycles. In [FG06, chapter 13], colour coding is used to obtain, for every polynomial time decidable class  $C$  of structures of bounded tree width, an fpt algorithm deciding the problem of whether there exists an embedding from  $A$  to  $B$ , where  $A$  is a structure from  $C$  and  $B$  an arbitrary structure.

### 2.1.2 Schöning’s Algorithm

Another famous application of randomness in computer science is Schöning’s Algorithm [Sch02] for deciding whether a given propositional formula  $\varphi$  in  $k$ -CNF with  $n$  variables is satisfiable or not. The algorithm works as follows: Pick any assignment  $a_0$  to the variables at random. If  $a_0$  satisfies  $\varphi$ , answer “yes”. Otherwise choose any clause which is not satisfied, and flip the assignment of one of the at most  $k$  variables occurring in this clause to get a new assignment  $a_1$ . Repeat this  $M$  times. If no satisfying assignment has been found, guess a completely new assignment and start a new round. If, after  $N$  rounds, no satisfying assignment has been found, answer “no”.

Increasing  $M$  and  $N$  in this algorithm increases the probability of hitting a satisfying assignment, if one exists; at the cost of increasing the running time. By a clever analysis of the probability of hitting a satisfying assignment in the course of this algorithm, one can show that there is a randomised algorithm with a running time of  $(2(k-1)/k)^n \cdot n^{O(1)}$

which finds a satisfying assignment with probability  $\geq 1/2$  if such an assignment exists.

This randomised algorithm has subsequently been derandomised, this first derandomisation is by Dantsin et al. [DGH<sup>+</sup>02] and has a running time of  $(2k/(k+1))^n \cdot n^{O(1)}$ . A more recent derandomisation is by Moser and Scheder [MS10], giving a running time of  $(2(k-1)/k + \epsilon)^n \cdot n^{O(1)}$  with arbitrarily small  $\epsilon > 0$ . A key ingredient of these derandomisations is the construction of a *covering code*, i.e., a set  $C \subseteq \{0,1\}^n$  such that for every  $x \in \{0,1\}^n$  there is a  $y \in C$  such that the Hamming distance between  $x$  and  $y$  is at most  $r$ , for some specified  $r$  which is called the distance of the code. Again, by standard arguments one can show that drawing  $N = N(r)$  strings from  $\{0,1\}^n$  independently and uniformly at random will result in a covering code with high probability.

## 2.2 Inapproximability of Weighted Monotone Circuit Satisfiability

In [AR01], Alekhnovich and Razborov conditionally proved that resolution is not automatisable, i.e., there is no algorithm which, given an unsatisfiable propositional formula in conjunctive normal form, produces a resolution refutation of this formula and runs in time polynomial in the length of the shortest such refutation. They used the slightly non-standard assumption that  $W[P]$  is not equal to randomised fpt. Equivalently, they assume there is no randomised algorithm running in time  $f(k) \cdot n^{O(1)}$  which solves the following problem:

$p$ -WSAT(CIRC<sup>+</sup>)

*Input:* A monotone circuit  $C$  with  $n$  input nodes, and some  $k \geq 1$   
*Parameter:*  $k$   
*Problem:* decide if  $C$  has a satisfying assignment of Hamming weight  $k$

While (non-randomised)  $FPT = W[P]$  has unpalatable consequences in classical complexity, making  $FPT \neq W[P]$  a fairly common assumption in parameterized complexity (cf. [FG06, sec. 3.3]), Alekhnovich and Razborov's assumption about randomised FPT is far less common and arises as an artefact of their proof technique. Namely, they show that an algorithm automatising resolution could be used to obtain an FPT-algorithm for the following promise problem with  $\delta = 2$ :

## 2.2 Inapproximability of Weighted Monotone Circuit Satisfiability

### $p$ - $\delta$ -GAP-WSAT(CIRC<sup>+</sup>)

<i>Input:</i>	A monotone circuit $C$ with $n$ input nodes, and some $k \geq 1$
<i>Parameter:</i>	$k$
<i>Promise:</i>	if $C$ has a satisfying assignment of Hamming weight $\delta k$ , it also has one of Hamming weight $k$
<i>Problem:</i>	decide if $C$ has a satisfying assignment of Hamming weight $k$

In general, we allow  $\delta$  to depend on  $k$ . Alekhovich and Razborov then prove that this promise problem with  $\delta = 2$  is W[P]-complete by reducing  $p$ -WSAT(CIRC<sup>+</sup>) to it. This amounts to introducing a *gap* as follows: Given a monotone circuit  $C$  and a  $k \geq 1$ , they construct a new monotone circuit  $C'$  and a new parameter  $k'$  such that:

- If  $C$  has a satisfying assignment of Hamming weight  $k$ , then  $C'$  has a satisfying assignment of Hamming weight  $k'$ .
- If  $C$  has no satisfying assignment of Hamming weight  $k$ , then  $C'$  has no satisfying assignment of Hamming weight  $2k'$ .
- There is some computable function  $f$  such that  $C'$  has size  $\leq f(k) \cdot |C|^{O(1)}$ , and  $k'$  is bounded by  $f(k)$ .

Their reduction is a randomised algorithm, which uses certain graphs which have an expansion property that a random graph has with high probability but for which they give no deterministic construction. In [EGG08] we give a deterministic construction of graphs with a somewhat weaker expansion property (but still sufficient for the purpose of the above reduction), thereby proving that resolution is not automatisable unless W[P] = FPT.

### 2.2.1 Details of the reduction

Recall our definition of Boolean circuits from section 1.4.3. The (*Hamming*) *weight* of an assignment  $a$  to the inputs of a circuit  $C$  is the number of 1-entries of  $a$ , and  $\min(C)$  is defined to be the minimum weight of a satisfying assignment. If  $\min(C) \leq k$  we say that  $C$  is *k-satisfiable*. A Boolean circuit is *monotone* if it does not contain any negation nodes. CIRC denotes the class of all Boolean circuits and CIRC<sup>+</sup> the class of all monotone Boolean circuits.

The main result of this section is the following lemma:

**Lemma 13.** *For any  $\delta > 1$ , there is an fpt many-one-reduction from  $p$ -WSAT(CIRC<sup>+</sup>) to  $p$ - $\delta$ -GAP-WSAT(CIRC<sup>+</sup>). That is, given  $k \in \mathbb{N}$ , and a monotone circuit  $C$  with  $n$  inputs, we can deterministically construct a circuit  $\pi(C, k, \delta)$  such that*

$$\min(C) \leq k \quad \Rightarrow \quad \min(\pi(C, k, \delta)) \leq \alpha(k, \delta) \quad (2.1)$$

## 2 Random and Pseudorandom Structures

and

$$\min(C) \geq k + 1 \quad \Rightarrow \quad \min(\pi(C, k, \delta)) \geq \delta \alpha(k, \delta). \quad (2.2)$$

Here,  $\alpha(k, \delta)$  depends only on  $k$  and  $\delta$ , and  $\pi(C, k, \delta)$  has size  $g(k)n|C|$  for some computable function  $g$  and can be computed by an fpt algorithm.

We start by constructing certain unbalanced bipartite graphs with good expansion properties, which we will need later.

**Lemma 14.** *For any  $\epsilon > 0$ , integer  $t \geq 2$ , and  $K_{\max} \in \mathbb{N}$ , we set*

$$d := \left\lceil \frac{K_{\max} \cdot (t - 1)}{2\epsilon} \right\rceil. \quad (2.3)$$

Then for any prime power  $q > d$  we can explicitly construct a bipartite graph  $G = (V, E)$  with left degree  $d$ , left vertex set  $L$  of size  $q^t$ , and right vertex set  $R$  of size  $dq$ , such that

$$\forall W \subseteq L, |W| \leq K_{\max} : |\Gamma(W)| \geq (1 - \epsilon)d|W|,$$

where  $\Gamma(W) = \{r \in R \mid E(v, r) \text{ for some } v \in W\}$  is the set of neighbours of  $W$ .

In other words,  $G$  is a  $(K_{\max}, \epsilon)$ -lossless expander: For small ( $\leq K_{\max}$  elements) sets of left vertices we have only very few collisions, resulting in nearly lossless expansion. Moreover, the left degree of our expander depends only on  $K_{\max}$  and  $\epsilon$ , not on  $q$ .

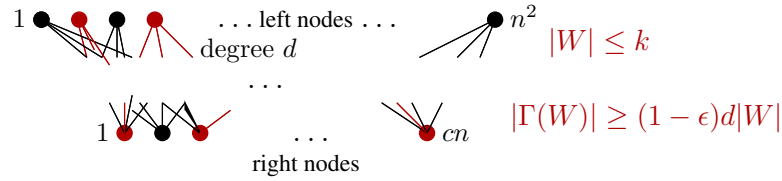


Figure 2.1: Bipartite lossless expanders.

*Proof of Lemma 14.* We give an explicit construction. Suppose that  $q > d$  is a prime power. For the vertex sets  $L$  and  $R$ , we choose

$$L := \mathbb{F}_q^t \quad \text{and} \quad R := [d] \times \mathbb{F}_q,$$

where  $[d] = \{1, \dots, d\}$  and  $\mathbb{F}_q$  is the Galois field with  $q$  elements. We pick vectors  $\mathbf{u}_1, \dots, \mathbf{u}_d \in \mathbb{F}_q^t$  such that any  $t$  of them are linearly independent, for example

$$\mathbf{u}_i := \left(1, x_i, \dots, x_i^{t-1}\right)^T,$$

where  $x_1, \dots, x_d$  are pairwise different elements of  $\mathbb{F}_q$ . We connect the vertex  $\mathbf{v} \in L$  to the vertices

$$(1, \mathbf{v}^T \mathbf{u}_1), \quad \dots, \quad (d, \mathbf{v}^T \mathbf{u}_d).$$

## 2.2 Inapproximability of Weighted Monotone Circuit Satisfiability

Notice that no two vertices of  $L$  can have more than  $t-1$  neighbours in common, because any  $t$  of the vectors  $u_1, \dots, u_d$  form a basis of  $\mathbb{F}_q^t$ . Therefore, if  $W \subseteq L$  has at most  $K_{\max}$  elements, then

$$\begin{aligned} |\Gamma(W)| &= \left| \bigcup_{\mathbf{v} \in W} \Gamma(\mathbf{v}) \right| \\ &\geq \sum_{\mathbf{v} \in W} |\Gamma(\mathbf{v})| - \sum_{\mathbf{v} \neq \mathbf{w} \in W} |\Gamma(\mathbf{v}) \cap \Gamma(\mathbf{w})| \\ &\geq d|W| - \binom{|W|}{2}(t-1) \\ &\geq d|W| \left( 1 - \frac{(t-1)|W|^2}{2d|W|} \right) \\ &\geq d|W|(1-\epsilon) \end{aligned}$$

by our choice of  $d$ . □

**Remark 15.** In Lemma 14 it is crucial that the left degree  $d$  does not depend on  $q$ . This is because we want the position of the gap in the circuit  $\pi(C, k, \delta)$  which we will construct in Lemma 13 to depend only on  $k$  and  $\delta$ , but not on  $n$ .

In [CRVW02], Capalbo et al. gave a construction of lossless expanders, but there the left degree grows polylogarithmically in  $L/R$ , the quotient of the number of left and right vertices, which is  $O(q)$  in our case. The benefit of their expanders is that  $K_{\max} = \epsilon L$  grows linear with the number of left vertices. The expanders constructed by Guruswami et al. [GUV07] are even more unbalanced than our expanders ( $R = \text{polylog}(L)$ ), but with a degree polylogarithmic in  $L$ .

These expanders can also be seen as error correcting codes or as a family of  $d$ -element subsets of  $[dq]$  such that any two subsets have small intersection. Nisan and Wigderson [NW88] constructed a system of  $q$ -element subsets of  $[q^2]$ , such that any two sets intersect in at most  $\log q$  elements. They essentially use Reed-Solomon-Codes over  $\mathbb{F}_q$ , the same construction is used, for example, in [BMRV00] to devise a randomised query scheme for storing subsets. Again, the size of the sets grows with  $q$ , so we can not use this construction here.

*Proof of Lemma 13.* We construct the circuit  $\pi(C, k, \delta)$  by starting with a copy of  $C$ , below which we add layers of copies of  $C$  as shown in Figure 2.2. Each layer achieves a certain gap amplification, while only increasing the number of inputs by a factor depending only on  $k$ . To be precise, the layers have the following properties:

- (a) Layer  $\ell$  is a monotone circuit with  $I_\ell$  inputs and  $O_\ell$  outputs, where

$$O_1 := n, \quad d_\ell O_\ell \leq I_\ell < 2d_\ell O_\ell, \quad O_{\ell+1} := I_\ell.$$

Here,  $d_\ell$  is a constant to be specified later which depends only on  $k$  and  $\ell$ . We will use the notation  $D_\ell := d_1 \cdot d_2 \cdots d_\ell$  for the product of the first  $\ell$  of these constants

## 2 Random and Pseudorandom Structures

(with  $D_0 := 1$ ).

- (b) If  $\min(C) \leq k$ , then for any set  $S$  of  $D_{\ell-1}k^\ell$  outputs of layer  $\ell$  there is an assignment of weight  $D_\ell k^{\ell+1}$  to the inputs of that layer such that (at least) all the outputs in  $S$  are satisfied.
- (c) If, on the other hand,  $\min(C) \geq k + 1$ , then there is no assignment of weight less than  $D_\ell(k^{\ell+1} + (\ell + 1)k^\ell)$  to the inputs of layer  $\ell$  which satisfies  $D_{\ell-1}(k^\ell + \ell k^{\ell-1})$  or more of the outputs of that layer.
- (d) For fixed  $k$ , the size of layer  $\ell$  as a circuit depends linearly on  $n \cdot |C|$  (so it is quadratic in the size of  $C$ ).

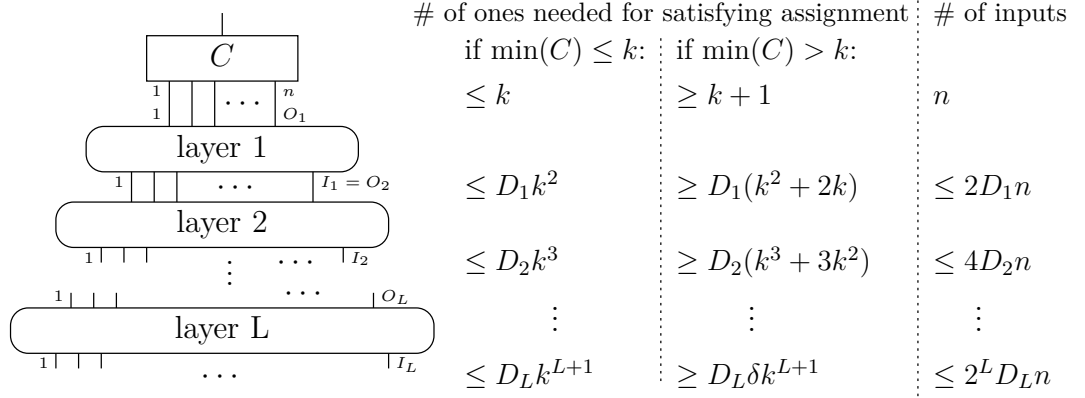


Figure 2.2: The overall structure of  $\pi(C, k, \delta)$

We choose

$$L := \lceil (\delta - 1)k - 1 \rceil$$

and see by descending down the layers using property (c) that if  $\min(C) \geq k + 1$  we need at least

$$D_L \cdot (k^{L+1} + (L + 1)k^L) \geq D_L \cdot \delta \cdot k^{L+1}$$

many ones to satisfy  $\pi(C, k, \delta)$ , while in the case  $\min(C) \leq k$  we need only  $D_L k^{L+1}$  many by property (b). Thus, both (2.1) and (2.2) are satisfied, with  $\alpha(k, \delta) := D_L k^{L+1}$ .

It remains to describe the construction of the individual layers (cf. Figure 2.3). Each of the  $O_\ell$  outputs of layer  $\ell$  is connected to a copy of  $C$ . These have a total of  $n \cdot O_\ell$  inputs. We let  $\tilde{O}_\ell$  be the least power of two greater than or equal to  $O_\ell$ , so that  $O_\ell \leq \tilde{O}_\ell < 2O_\ell$ . By induction, property (a) implies that  $O_\ell \geq n$  for all  $\ell$ , so we get

$$n \cdot O_\ell \leq O_\ell^2 \leq \tilde{O}_\ell^2.$$

We use Lemma 14 with parameters

$$t := 2,$$

## 2.2 Inapproximability of Weighted Monotone Circuit Satisfiability

$$K_{\max} := K_\ell := D_{\ell-1}(k^\ell + \ell k^{\ell-1})(k+1),$$

$$\epsilon := \epsilon_\ell := \frac{\ell k^{\ell-1}}{(k^\ell + \ell k^{\ell-1})(k+1)}$$

to construct a bipartite expander with  $\tilde{O}_\ell^2$  left vertices, left degree  $d_\ell$  defined as in (2.3) and  $d_\ell \tilde{O}_\ell$  right vertices. For each of the right vertices we introduce an input of layer  $\ell$ . We view the  $n \cdot O_\ell$  inputs of the copies of  $C$  as (a subset of the) left vertices of this expander and connect each of them to the conjunction of  $d_\ell$  of the inputs of layer  $\ell$ .

This construction obviously satisfies properties (a) and (d). To see that (b) also holds, we assume that  $\min(C) \leq k$ . Then there exists a satisfying assignment of weight  $\leq k$  for  $C$ , so if we are given a subset  $S$  of  $D_{\ell-1}k^\ell$  outputs of layer  $\ell$  it suffices to satisfy  $k \cdot D_{\ell-1}k^\ell$  many of the and-gates in that layer. But these are connected to at most  $d_\ell$  inputs each, so there is an assignment of weight  $D_\ell k^{\ell+1}$  to the inputs of layer  $\ell$  such that all outputs in  $S$  are satisfied.

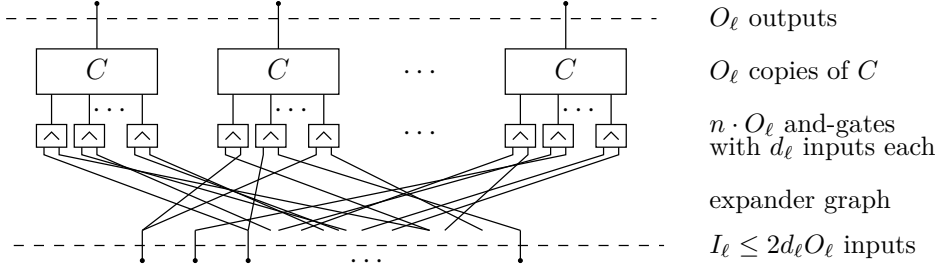


Figure 2.3: Layer  $\ell$  of  $\pi(C, k, \delta)$

If, on the other hand, there is no assignment of weight  $\leq k$  which satisfies  $C$ , then for  $D_{\ell-1}(k^\ell + \ell k^{\ell-1})$  of the output gates of layer  $\ell$  to be satisfied, at least

$$(k+1)D_{\ell-1}(k^\ell + \ell k^{\ell-1}) = K_\ell$$

many of the and-gates in that layer must be satisfied. By the expansion property of our wiring, any set of  $K_\ell$  and-gates is connected to at least

$$(1 - \epsilon_\ell)d_\ell K_\ell = D_\ell(k^{\ell+1} + (\ell+1)k^\ell)$$

many inputs of the layer, therefore no satisfying assignment of weight less than this number can exist and (c) is proved.  $\square$

### 2.2.2 Parameterized Inapproximability

Parameterized approximability is a relaxed notion of classical approximability. Intuitively, an fpt approximation algorithm is an algorithm whose running time is fpt for the parameter “cost of the solution” and whose approximation ratio only depends on the parameter and not on the size of the input. Hence every polynomial time approximation

algorithm with constant approximation ratio is an fpt approximation algorithm, but an approximation algorithm with approximation ratio  $\log n$ , where  $n$  denotes the input size, is not. We will only give the definitions related to fpt approximability for minimisation problems, but it is straightforward to adapt them to maximisation problems.

**Definition 16.** Let  $\rho : \mathbb{N} \rightarrow \mathbb{R}_{>1}$  be a computable function. An *fpt approximation algorithm* for an NP minimisation problem  $O$  (over some alphabet  $\Sigma$ ) with approximation ratio  $\rho$  is an algorithm  $\mathbb{A}$  with the following properties:

1.  $\mathbb{A}$  expects inputs  $(x, k) \in \Sigma^* \times \mathbb{N}$ . For every input  $(x, k) \in \Sigma^* \times \mathbb{N}$  such that there exists a solution for  $x$  of cost at most  $k$ , the algorithm  $\mathbb{A}$  computes a solution for  $x$  of cost at most  $k \cdot \rho(k)$ . For inputs  $(x, k) \in \Sigma^* \times \mathbb{N}$  without solution of cost at most  $k$ , the output of  $\mathbb{A}$  can be arbitrary.
2. There exists a computable function  $f$  such that the running time of  $\mathbb{A}$  on input  $(x, k)$  is bounded by  $f(k) \cdot |x|^{O(1)}$ .

In our inapproximability results, we will work with a weaker notion of approximability where an algorithm is only required to compute the cost of an optimal solution rather than an actual solution; this notion was called *cost approximability* in [CGG06]. It will be convenient to define cost approximability in terms of certain decision problems associated with the optimisation problems. Instances of the *standard decision problem* associated with a minimisation problem  $O$  are pairs  $(x, k)$ , where  $x$  is an instance of  $O$  and  $k$  a natural number, and the problem is to decide if  $\min(x) \leq k$ . Taking  $k$  as parameter, we obtain the *standard parameterization* of the minimisation problem  $O$ . We define cost approximability in terms of a gap version of the standard decision problem:

**Definition 17.** Let  $O$  be an NP minimisation problem over the alphabet  $\Sigma$  and let  $\rho : \mathbb{N} \rightarrow \mathbb{R}_{>1}$  be a computable function. Then a decision algorithm  $\mathbb{A}$  is an *fpt cost approximation algorithm* for  $O$  with *approximation ratio*  $\rho$  if it is an fpt algorithm satisfying the following conditions for all inputs  $(x, k) \in \Sigma^* \times \mathbb{N}$  such that there exists at least one solution for  $x$ :

1. If  $k \geq \min(x) \cdot \rho(\min(x))$ , then  $\mathbb{A}$  accepts  $(x, k)$ .
2. If  $k < \min(x)$ , then  $\mathbb{A}$  rejects  $(x, k)$ .

(For instances  $x$  with no valid solution, the algorithm  $\mathbb{A}$  can be assumed to reject  $(x, k)$  for all  $k \in \mathbb{N}$ .)

It is easy to see that fpt approximability implies fpt cost approximability with the same ratio (cf. [CGG06]).

**Theorem 18.** For  $\text{MIN-WSAT}(\text{CIRC}^+)$ , there exists no fpt cost approximation algorithm with ratio

$$\rho(k) = \exp(\log^\gamma k), \quad \text{where } \gamma < \frac{\log 2}{\log 6} \approx 0.387,$$

unless  $\text{W[P]} = \text{FPT}$ .



## 2.2 Inapproximability of Weighted Monotone Circuit Satisfiability

*Proof.* We use Lemma 13 to reduce the standard parameterization  $p\text{-WSAT}(\text{CIRC}^+)$  to its approximation variant.

We first show that if there were a constant fpt cost approximation algorithm for  $\text{MIN-WSAT}(\text{CIRC}^+)$ , this could be used to solve  $p\text{-WSAT}(\text{CIRC}^+)$  by an fpt algorithm. Say  $\mathbb{A}$  is such an algorithm with constant approximation ratio  $\rho(k) = c$  for all  $k$ . Given a circuit  $C$  and a parameter  $k$ , we wish to decide whether or not  $C$  is  $k$ -satisfiable. We use Lemma 13 with  $\delta = c + 1$  to obtain a circuit  $C'$ , and run algorithm  $\mathbb{A}$  on  $(C', c \cdot \alpha)$ , where  $\alpha = \alpha(k, \delta)$  is as in the lemma (note that it can easily be computed from  $\delta$  and  $k$ ).

Now, if  $\min(C) \leq k$ , then  $\min(C') \leq \alpha$ , so  $c \cdot \min(C') \leq c \cdot \alpha$  and the algorithm accepts. If, on the other hand,  $\min(C) > k$ , then  $\min(C') \geq \delta \cdot \alpha > c \cdot \alpha$ , so in this case the algorithm rejects. In summary, we have gained an fpt algorithm for  $p\text{-WSAT}(\text{CIRC}^+)$ . Because this problem is  $\text{W[P]}$ -hard, it follows that  $\text{W[P]} = \text{FPT}$ .

We sharpen this result, starting from an fpt cost approximation algorithm for  $\text{MIN-WSAT}(\text{CIRC}^+)$  with approximation ratio

$$\rho(k) = \exp(\log^\gamma k), \quad \gamma < \frac{\log 2}{\log 6}.$$

Given a circuit  $C$  and a parameter  $k$ , we seek to find a circuit  $C''$  such that

$$\min(C'') \begin{cases} \leq \alpha & \text{if } \min(C) \leq k, \\ > \rho(\alpha) \cdot \alpha & \text{if } \min(C) > k. \end{cases}$$

The problem is that the construction of Lemma 13 does not only increase the gap, but at the same time also increases its position. Here we need a gap size that grows with the position of the gap.

We choose  $\beta$  such that

$$6^\gamma < \beta < 6^{\log 2 / \log 6} = 2$$

and set

$$\delta_0 = 2^{\frac{1}{2-\beta}}.$$

Then we use Lemma 13 to obtain a circuit  $C'$  with a gap at position  $\alpha_0$  of relative size  $\delta_0$  for some  $\alpha_0$ . This means that either there is a satisfying assignment of weight at most  $\alpha_0$ , or any satisfying assignment has weight at least  $\delta_0 \cdot \alpha_0$ . The overall structure of  $C''$  is similar to the construction of Lemma 13, see Figure 2.4. Suppose that before level  $\ell$  of the construction, we have a gap of relative size  $\delta_{\ell-1}$  at position  $\alpha_{\ell-1}$ . As before, each layer looks like in Figure 2.3, with  $C'$  instead of  $C$  and the parameters for the expander chosen as follows:

$$\begin{aligned} \epsilon_\ell &:= \frac{1}{2}, \\ K_\ell &:= (\delta_{\ell-1} \cdot \alpha_{\ell-1})^2 \quad \text{and} \\ t &= 2. \end{aligned}$$

## 2 Random and Pseudorandom Structures

By (2.3), the expander therefore has left degree  $d_\ell = K_\ell$ . If  $\delta_{\ell-1} > \rho(\alpha_{\ell-1})$ , we would not need another layer, so in particular we may assume that  $\delta_{\ell-1} \leq \alpha_{\ell-1}$ . Layer  $\ell$  moves the gap to

$$\begin{aligned}\alpha_\ell &= \alpha_{\ell-1}^2 \cdot d_\ell \\ &= \alpha_{\ell-1}^2 \cdot (\delta_{\ell-1} \alpha_{\ell-1})^2 \\ &\leq \alpha_{\ell-1}^6,\end{aligned}$$

while increasing its size to

$$\begin{aligned}\delta_\ell &\geq (1 - \epsilon_\ell) \delta_{\ell-1}^2 \\ &= \delta_{\ell-1}^\beta \cdot \left(\frac{1}{2} \delta_{\ell-1}^{2-\beta}\right) \\ &\geq \delta_{\ell-1}^\beta,\end{aligned}$$

where the last inequality follows from  $\delta_{\ell-1} \geq \delta_0$ , which is easily seen by induction, and our choice of  $\delta_0$ .

We see that after layer  $\ell$  the gap is at position

$$\alpha_\ell \leq \alpha_0^{6^\ell}$$

and has relative size

$$\delta_\ell \geq \delta_0^{\beta^\ell}.$$

The total number  $L$  of layers must be big enough to satisfy

$$\delta_0^{\beta^L} > \rho(\alpha_0^{6^L}) = \exp\left(\beta'^L \cdot \log^\gamma \alpha_0\right),$$

where  $\beta' := 6^\gamma < \beta$ , and thus we need

$$\left(\frac{\beta}{\beta'}\right)^L > \frac{\log^\gamma \alpha_0}{\log \delta_0},$$

and set

$$L := \left\lceil \frac{\log \log^\gamma \alpha_0 - \log \log \delta_0}{\log \beta - \log \beta'} \right\rceil,$$

which for fixed  $\gamma$  and  $\beta$  depends only on  $k$ . The size of the resulting circuit can be estimated as in Lemma 13.  $\square$

Using a notion of reduction called *ftp gap-preserving reduction*, similar non-approximability results can be obtained for other approximation problems as well. In [EGG08], we show that the parameterised minimisation problems Min-Chain-Reaction-Closure, Min- $t$ -Threshold-Starting-Set, Min-Generating-Set, Min-Axiom-Set, Min-Degree-3-Subgraph-Annihilator, Min-Linear-Inequality-Deletion, Min-Induced-SAT(3CNF) and Min-Induced-SAT are not ftp cost approxiable either, with similar ratios  $\rho$  as above.

## 2.3 Inapproximability of the Minmax Value in Three Player Games

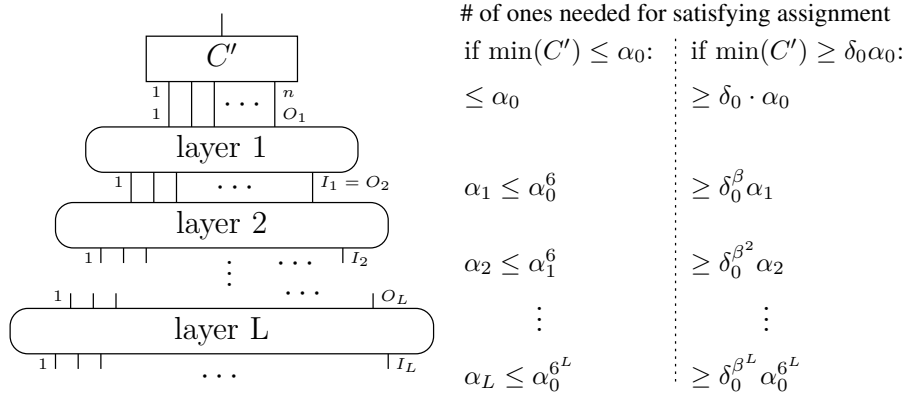


Figure 2.4: Refined gap amplification.

## 2.3 Inapproximability of the Minmax Value in Three Player Games

In this section, we consider the problem of approximating the minmax value of a multiplayer game in strategic form. We argue that in 3-player games with 0-1 payoffs, approximating the minmax value within an additive constant smaller than  $\phi/2$ , where  $\phi \approx 0.382$ , is not possible by a polynomial time algorithm. This is based on assuming hardness of a version of the so-called planted clique problem in random graphs, namely that of *detecting* a planted clique. Our results are stated as reductions from a promise graph problem to the problem of approximating the minmax value, and we use the detection problem for planted cliques to argue for its hardness. We present two reductions: a randomised many-one reduction and a deterministic Turing reduction. The latter, which may be seen as a derandomisation of the former, may be used to argue for hardness of approximating the minmax value based on a hardness assumption about *deterministic* algorithms.

### 2.3.1 The Minmax Value in Three Player Games

We consider games in strategic form between 3 players. These are given by a finite strategy space for each player,  $S_1, S_2$ , and  $S_3$  (also called the *pure strategies*), together with utility functions  $u_1, u_2, u_3 : S_1 \times S_2 \times S_3 \rightarrow \mathbb{R}$ . We will identify the strategy spaces with the sets  $[n_1], [n_2]$ , and  $[n_3]$ , where  $n_i = |S_i|$ . We shall refer to this as an  $n_1 \times n_2 \times n_3$  game.

Let  $\Delta_1, \Delta_2$ , and  $\Delta_3$  be the sets of probability distributions over  $S_1, S_2$ , and  $S_3$  respectively; these are also called *mixed strategies*. The minmax value (also known as the threat value) for Player 1 is given by:

$$\min_{(\sigma_2, \sigma_3) \in \Delta_2 \times \Delta_3} \max_{\sigma_1 \in \Delta_1} \mathbb{E}_{a_i \sim \sigma_i} [u_1(a_1, a_2, a_3)]$$

## 2 Random and Pseudorandom Structures

A strategy profile  $(\sigma_2, \sigma_3)$  for Player 2 and Player 3 for which this value is obtained is called an optimal minmax profile. It is not hard to see that Player 1 may always obtain the maximum by a pure strategy, i.e., the minmax value is equal to:

$$\min_{(\sigma_2, \sigma_3) \in \Delta_2 \times \Delta_3} \max_{a_1 \in S_1} \mathbb{E}_{a_2 \sim \sigma_2, a_3 \sim \sigma_3} [u_1(a_1, a_2, a_3)] \quad (2.4)$$

To the minmax value, only the utilities for Player 1 are relevant, and we collect these in matrices  $A_i$  of size  $n_2 \times n_3$ , one for each pure strategy  $i \in S_1$ , by setting  $a_{j,k}^{(i)} = u_1(i, j, k)$ . In this notation, if Player 1 plays the pure strategy  $i$  and Player 2 and Player 3 play by mixed strategies  $x$  and  $y$ , the expected payoff to Player 1 is given by  $x^\top A^{(i)} y$ .

The corresponding notion of minmax value in finite two-player games is a fundamental notion of game theory. Minmax values have been studied much less in multi-player games, but they are arguably also of fundamental interest. In particular the minmax value of such games are crucial for the statements as well as proofs of the so-called *folk theorems* that characterise the Nash equilibria of *repeated games*. The problem of *computing* the minmax value of a multi-player game was first considered only recently by Borgs et al. [BCI<sup>+</sup>10], exactly in the context of studying computational aspects of the folk theorem. In particular they show that approximating the minmax value of a 3 player game within a specific inverse polynomial additive error is NP hard.

Here, to be able to talk meaningfully about approximation within an additive error, we assume that all payoffs have been *normalised* to be in the interval between 0 and 1. The question of approximating the minmax value was considered further by Hansen et al. [HHMS08]. Using a “padding” construction it was observed that the NP hardness result of Borgs et al. extends to any inverse polynomial additive error. This was complemented by a quasipolynomial approximation algorithm obtaining an approximation to within an arbitrary additive  $\epsilon > 0$ . This was obtained by a result of Lipton and Young [LY94], stating that in an  $n \times n$  matrix game with payoffs between 0 and 1, each player can guarantee a payoff within any  $\epsilon > 0$  of the value of the game using strategies that simply consist of a uniform choice from a multiset of  $\lceil \ln n / (2\epsilon^2) \rceil$  pure strategies. We summarise these results by the following theorem.

**Theorem 19** ([BCI<sup>+</sup>10, HHMS08]). *For any constant  $\epsilon > 0$  it is NP hard to approximate the minmax value of an  $n \times n \times n$  game with 0-1 payoffs within additive error  $1/n^\epsilon$ . On the other hand, there is an algorithm that, given  $\epsilon > 0$  and a  $n \times n \times n$  game with payoffs between 0 and 1, approximates the minmax value from above with additive error at most  $\epsilon$  in time  $n^{O(\log(n)/\epsilon^2)}$ .*

This naturally raises the question of whether it is possible to approximate the minmax value within any constant  $\epsilon > 0$  in polynomial time, or even whether it is possible to approximate the minmax value within *some* nontrivial additive constant  $0 < \epsilon < 1/2$  in polynomial time. Due to the quasipolynomial time algorithm above, it is unlikely that the theory of NP completeness can shed light on this question.

A similar situation is present for the problem of computing a Nash equilibrium in two player bimatrix games. Celebrated recent results [DGP09, CDT09] show that this

problem is complete for the complexity class PPAD. On the other hand, several works provide algorithms for computing an  $\epsilon$ -Nash equilibrium. An  $\epsilon$ -Nash equilibrium in a  $n \times n$  bimatrix game with payoffs between 0 and 1 can be computed in time  $n^{O(\log(n)/\epsilon^2)}$  [LMM03], by an algorithm similar to the one described above for the minmax value. As for polynomial time algorithms, several algorithms have been devised for decreasing the additive error  $\epsilon$  [KPS09, DMP06, DMP07, BBM10, TS08]. Currently, the best such algorithm achieves  $\epsilon = 0.3393$  [TS08]. How well a Nash equilibrium can be approximated in the sense of  $\epsilon$ -Nash equilibria is a major open question. Having a polynomial time algorithm, polynomial also in  $1/\epsilon$ , or in other words having a fully polynomial time approximation scheme (FPTAS), would imply that every problem in the class PPAD would be solvable in polynomial time [CDT09]. Currently there is no evidence for or against the existence of a polynomial time algorithm for any fixed  $\epsilon > 0$ , or in other words a polynomial time approximation scheme (PTAS) for computing  $\epsilon$ -Nash equilibria.

### The planted clique problem

Our result depends on assuming hardness of the so-called planted clique problem. Let  $G_{n,p}$  denote the distribution of Erdős-Rényi random graphs on  $n$  vertices where each potential edge is included in the graph independently at random with probability  $p$ . Most frequently the case of  $p = 1/2$  is considered, but we will be interested in having  $p > 0$  be a small constant. It is well known that in almost every graph from  $G_{n,p}$  the largest clique is of size  $2 \log_{1/p} n - O(\log \log n)$  [Bol01]. The hidden clique problem is defined using the distribution  $G_{n,p,k}$  [Jer92, Kuč95] of graphs on  $n$  vertices defined as follows: A graph  $G$  is picked according to  $G_{n,p}$ , then a set of  $k$  vertices are chosen uniformly at random and connected to form a clique. Thus apart from the planted  $k$ -clique the graph is completely random. We can now consider the computational problem of finding the planted  $k$ -clique in a graph chosen from  $G_{n,p,k}$ . Note that when the parameter  $k$  is significantly larger than  $2 \log_{1/p} n$ , the planted clique is with high probability the unique maximum clique in the graph.

The planted clique problem is known as a hard computational problem. Indeed the current best polynomial time algorithms for solving the planted clique problem [AKS98, FK00] are only known to work when  $k = \Omega(\sqrt{n})$ . We may compare this with the observation due to Kučera [Kuč95] that for  $k \geq C\sqrt{n \log n}$  when  $C$  is a suitably large constant, the vertices of the clique would almost surely be the vertices of largest degree, and hence easy to find. The planted clique problem has also been proposed as a basis for a cryptographic one-way function [JP00]. For this application, however, the size of the planted clique is  $k = (1 + \epsilon) \log_{1/p} n$ , which is smaller than the expected size of the largest clique.

### 2.3.2 Our Results

We show a relationship between the task of approximating the minmax value in a 3-player game and the task of detecting whether a random graph contains a large planted clique. Our result builds heavily on the work of Hazan and Krauthgamer in [HK11] and

## 2 Random and Pseudorandom Structures

[MV09] (which we describe in the next section).

In our results we prove hardness of approximating the minmax value, and aim to obtain a conclusion as strong as possible, while maintaining a reasonable assumption.

We will actually state our results using the following promise graph problem, parameterised by numbers  $0 < c_1 < c_2$  and  $\eta > 0$ . Let  $G = (V_1, V_2, E)$  be a bipartite graph. For  $S \subseteq V_1, T \subseteq V_2$  the *density* of the subgraph induced by  $S$  and  $T$  is given by

$$d(S, T) = \frac{|E(S, T)|}{|S||T|}.$$

Note that if we let  $A$  denote the adjacency matrix of  $G$  and let  $u_S$  and  $u_T$  be the probability vectors that are uniform on the sets  $S$  and  $T$ , then we have  $d(S, T) = u_S^T A u_T$ .

### GAP DENSE BIPARTITE SUBGRAPH (GAP-DBS)

*Input:* Bipartite graph  $G = (V_1, V_2, E)$ ,  $|V_1| = |V_2| = n$

*Promise:* Either

- (i) For all  $S \subseteq V_1, T \subseteq V_2, |S| = |T| = c_1 \ln n$ , it holds that  $d(S, T) \leq \eta$ , or
- (ii) There exist  $S \subseteq V_1, T \subseteq V_2, |S| = |T| = c_2 \ln n$ , such that  $d(S, T) \geq 1 - \eta$ .

*Problem:* Decide which of these is the case

We also introduce the following gap problem for the minmax value of 3-player games with 0-1 payoffs, parameterized by numbers  $0 \leq \alpha < \beta \leq 1$

### GAP 3 PLAYER MINMAX (GAP-MINMAX)

*Input:*  $n \times n \times n$  game  $G$  with 0-1 payoffs

*Promise:* Either

- (i) The minmax value for Player 1 in  $G$  is at most  $\alpha$ , or
- (ii) The minmax value for Player 1 in  $G$  is at least  $\beta$ .

*Problem:* Decide which of these is the case

We are now ready to state our results.

**Theorem 20.** *There exist reductions from the Gap-DBS problem to the Gap-minmax problem as follows.*

1. For every  $\eta < 0.1$  and  $0 < c_1 < c_2$  satisfying

$$\frac{c_2}{c_1} > \frac{2 \ln(1/\eta)}{(1 - \eta)\eta^2}$$

### 2.3 Inapproximability of the Minmax Value in Three Player Games

there is a randomised many-one reduction from the Gap-DBS problem to the Gap-minmax problem with parameters  $(\alpha, \beta) = (\eta, \phi - \eta/5)$ .

2. For every  $\eta < 0.1$  and  $0 < c_1 < c_2$  satisfying

$$\frac{c_2}{c_1} > \frac{1}{\eta}$$

there is a deterministic Turing reduction from the Gap-DBS problem to the Gap-minmax problem with parameters  $(\alpha, \beta) = (\eta, \phi - \eta/5)$ .

We prove the two parts of this theorems as two separate theorems, stated as Theorem 22 and Theorem 26. In both cases,  $\phi \approx 0.382$  denotes 1 minus the conjugate golden ratio. Interestingly, this constant has previously turned up as the additive error  $\phi + \delta$ , for arbitrary  $\delta > 0$ , obtained by an approximation algorithm for computing  $\epsilon$ -Nash equilibria [DMP07].

One can view the second reduction in Theorem 20 as a derandomisation of the first reduction in Theorem 20. However, this derandomisation comes at the cost of turning the many-one reduction into a Turing reduction. On the other hand the required ratio between  $c_1$  and  $c_2$  is much smaller.

We will use the planted clique problem to argue that the Gap-DBS is hard for certain settings of parameters  $(c_1, c_2, \eta)$ . For this we use similar arguments as in [HK11, MV09]. Given a graph  $H$  that is an input to the planted clique detection problem, we let  $A$  be the adjacency matrix of  $H$  and let  $G$  be the bipartite graph that also has  $A$  as adjacency matrix. We wish to have the following property: If  $H$  was chosen from  $G_{n,p}$ , then with high probability then  $G$  belongs to case (i) of the Gap-DBS problem, and if  $H$  was instead chosen from  $G_{n,p,k}$  then with high probability  $G$  belongs to case (ii) of the Gap-DBS problem.

We will set the parameters to achieve this as follows. First we fix  $\eta > 0$  to a desired value based on how close to  $\phi$  we wish to have the gap in the Gap-minmax problem. Then we choose  $p > 0$  and  $c_1 > 0$  in order to ensure graphs from  $G_{n,p}$  end up as case (i). The choice of reduction we wish to use from Theorem 20, then dictates a choice for  $c_2$ , and we let  $k = c_2 \ln n$ . Note that this automatically ensures that graphs from  $G_{n,p,k}$  ends up as case (ii), since if  $S$  is the set of  $k$  nodes where the clique is placed, we have  $d(S, S) = 1 - 1/k$ . In the following lemma we state the fact that the parameters  $p$  and  $c_1$  can be chosen as required.

**Lemma 21.** *Let  $\eta > 0$  be arbitrary. Then there exists a choice of  $p > 0$  and  $c_1 > 0$  such that with high probability a graph  $G = (V, E)$  chosen from  $G_{n,p}$  satisfies the following: Let  $A$  be the adjacency matrix of  $G$ , let  $S, T \subseteq V$  be of size  $|S| = |T| = c_1 \ln n$ . Then  $u_S^T A u_T \leq \eta$ , where  $u_S$  and  $u_T$  are probability vectors uniform on  $S$  and  $T$ .*

*Proof.* First, consider fixed sets  $S, T \subseteq V$  of size  $|S| = |T| = c_1 \ln n$ . We will estimate the probability that  $u_S^T A u_T \leq \eta$ , and after that take a union bound over all such sets  $S$  and  $T$ .

## 2 Random and Pseudorandom Structures

The number of potential edges between  $S$  and  $T$  in  $G$  is exactly given by

$$\ell = |S||T| - \binom{|S \cap T|}{2} - |S \cap T|,$$

because the  $|S||T|$ -term counts edges within  $S \cap T$  twice and also counts loops in that set. Thus  $\ell \geq (c_1 \ln n)^2/3$  for large enough  $n$ . Letting  $X$  be a random variable denoting the number of edges between  $S$  and  $T$  we have  $\mathbb{E}[X] = p\ell$ . Note that if  $X \leq \eta/2 \cdot \ell$  then we have

$$u_S^\top A u_T \leq \eta \cdot \frac{\ell}{|S||T|} \leq \eta,$$

because each edge contributes at most  $2(|S||T|)^{-1}$ . We set  $p = \eta/(2e)$  and apply Chernoff's bound for the upper tail (cf. Theorem 1) with  $\delta = e - 1$  to get

$$\begin{aligned} \mathbb{P}(X > \eta/2 \cdot \ell) &= \mathbb{P}(X > p\ell) \\ &= \mathbb{P}[X > e\mathbb{E}[X]] \\ &< \left( \frac{e^{e-1}}{e^e} \right)^{\mathbb{E}[X]} \\ &= \exp(-\mathbb{E}[X]) \\ &= \exp(-p\ell) \\ &\leq \exp\left(-\frac{p}{3} \cdot (c_1 \ln n)^2\right). \end{aligned}$$

We then wish to take a union over all choices of sets  $S$  and  $T$ . We have at most

$$\binom{n}{c_1 \ln n}^2 \leq \exp(2c_1(\ln n)^2)$$

such sets. We can thus obtain the statement of the lemma, by letting  $c_1 > 6/p = 12e/\eta$ .  $\square$

### 2.3.3 Related Work

The problem of computing a Nash equilibrium in a bimatrix is PPAD complete. However, there are many different properties such that asking for a Nash equilibrium that satisfies the property is an NP hard problem [CS03, GZ89]. In particular it is NP hard to compute a Nash equilibrium maximizing the social welfare.

Hazan and Krauthgamer [HK11], motivated by the question of whether there is a PTAS for computing  $\epsilon$ -Nash equilibria, considered the question of computing an  $\epsilon$ -best  $\epsilon$ -Nash equilibrium. An  $\epsilon$ -best  $\epsilon$ -Nash equilibrium is an  $\epsilon$ -Nash equilibrium whose social welfare is no less than the maximal social welfare achievable by a Nash equilibrium, minus  $\epsilon$ . Hazan and Krauthgamer gave a randomised polynomial time reduction from the



## 2.3 Inapproximability of the Minmax Value in Three Player Games

planted clique problem to the problem of computing an  $\epsilon$ -best  $\epsilon$ -Nash equilibrium. More precisely, they show there are constants  $\epsilon, c > 0$  such that if there is a polynomial time algorithm that computes in a two-player bimatrix game an  $\epsilon$ -best  $\epsilon$ -Nash equilibrium, then there is a randomised polynomial time algorithm that solves the planted clique problem in  $G_{n,1/2}$  for  $k = c \log_2 n$  with high probability.

This result was sharpened by Minder and Vilenchik [MV09], who made the constant  $c$  smaller. In particular they obtain  $c = 3 + \delta$ , for arbitrary  $\delta > 0$  (here  $\delta > 0$  dictates an upper bound on  $\epsilon$ ), and for the similar problem of detecting a planted clique they obtain  $c = 2 + \delta$ . Essentially the goal of Minder and Vilenchik was the opposite of ours. Namely, viewing their result as arguing for hardness, their goal was to obtain an assumption as weak as possible, while maintaining a nontrivial conclusion.

### 2.3.4 The randomised reduction

In this section we present a randomised reduction from Gap-DBS to minmax-value in three player games. To be precise, we prove the following result:

**Theorem 22.** *Let  $0 < \eta < 0.1$  and  $0 < c_1 < c_2$  and such that*

$$\frac{c_2}{c_1} > \frac{2 \ln(1/\eta)}{(1 - \eta)\eta^2}$$

*Then there is a randomised polynomial time many-one reduction which, given as input the adjacency matrix  $A \in \{0, 1\}^{n \times n}$  of a bipartite graph  $G$ , outputs a three-player game  $G_A$  such that with high probability*

- *if there are subsets  $S, T \subseteq [n]$  of size at least  $c_2 \ln n$  such that  $d(S, T) \geq 1 - \eta$ , then*

$$\text{minmax}_1 G_A \leq \eta$$

- *if  $d(S, T) < \eta$  for every  $S, T \subseteq [n]$  of size at least  $c_1 \ln n$ , then*

$$\text{minmax}_1 G_A > \phi - \frac{\eta}{5},$$

where

$$\phi = \frac{3 - \sqrt{5}}{2} \approx 0.382$$

*is the smaller of the two roots of the polynomial  $x^2 - 3x + 1 = 0$ .*

We will need the following lemma:

**Lemma 23.** *Let  $0 < \delta < 1$ , and  $k_1 = c_1 \ln n$ ,  $k_2 = c_2 \ln n$ , where  $0 < c_1 < c_2$  satisfy*

$$c_2 > \frac{2 \ln(1/\delta)}{(1 - \delta)\delta^2} \cdot c_1$$

*Let  $D \subseteq [n]$  be a fixed subset of size  $|D| = k_2$ . Then there is a constant  $c$  such that if we choose at random  $m = n^c$  subsets  $S_1, \dots, S_m \subseteq [n]$ , by letting  $j \in S_i$  with probability*

## 2 Random and Pseudorandom Structures

$1 - \delta$ , independently for every  $i$  and  $j$ , with probability at least  $1 - n^{-\Omega(1)}$  the sets satisfy the following properties.

- (a) For all  $i$ ,  $|S_i \cap D| \geq (1 - \delta)^2 k_2$ .
- (b) For every set  $S \subseteq [n]$  of size  $|S| = k_1$ , there exists  $i$  such that  $S_i \cap S = \emptyset$ .

*Proof.* By assumption we can pick  $c$  such that

$$c_1 \cdot \ln(1/\delta) < c < \frac{(1 - \delta)\delta^2}{2} \cdot c_2 .$$

We first prove property (a) holds with the claimed probability. We have  $\mathbb{E}[|S_i \cap D|] = (1 - \delta)k_2$ . By the Chernoff bound for the lower tail (cf. Theorem 1), we have

$$\mathbb{P}[|S_i \cap D| < (1 - \delta)^2 k_2] < \exp(-(1 - \delta)\delta^2 k_2 / 2) = n^{-\frac{(1 - \delta)\delta^2}{2} \cdot c_2}$$

Hence

$$\mathbb{P}[\exists i : |S_i \cap D| < (1 - \delta)^2 k_2] < m \cdot n^{-\frac{(1 - \delta)\delta^2}{2} \cdot c_2} = n^{-\Omega(1)} .$$

We next prove that property (b) also holds with the claimed probability. Consider  $S \subseteq [n]$  of size  $|S| = k_1$ . Then  $\mathbb{P}[S_i \cap S \neq \emptyset] = 1 - \delta^{k_1}$ , and

$$\begin{aligned} \mathbb{P}[S_i \cap S \neq \emptyset \text{ for all } i] &= (1 - \delta^{k_1})^m \\ &< \exp(-\delta^{k_1} m) \\ &= \exp(-n^{c - c_1 \ln(1/\delta)}). \end{aligned}$$

Hence

$$\begin{aligned} \mathbb{P}[\exists S \subseteq [n], |S| = k_1 : \forall i : S_i \cap S \neq \emptyset] &< \binom{n}{k_1} \exp(-\delta^{k_1} m) \\ &\leq \exp(c_1 \ln^2(n) - n^{c - c_1 \ln(1/\delta)}) \\ &< \exp(-n^{\Omega(1)}) . \end{aligned}$$

□

*Proof of Thm. 22.* We use Lemma 23 with  $c_1$  and  $c_2$  as in the problem description and  $\delta = 1 - \sqrt{1 - \eta} = 1 - \eta/2 + O(\eta^2)$ . Let  $m$  be as in the lemma. The reduction first guesses  $2m$  subsets  $S_1^{(r)}, \dots, S_m^{(r)}, S_1^{(c)}, \dots, S_m^{(c)}$  at random as in the lemma. It then outputs a 3-player game  $G_A$  as follows:

- Players 2 and 3 have  $n$  strategies each.
- Player 1 has  $2m + 1$  strategies denoted by  $b, r_1, \dots, r_m$ , and  $s_1, \dots, s_m$ . The corresponding payoff matrices for player 1 are  $B = 1 - A$ ,  $R^{(k)}$  and  $C^{(k)}$  for

### 2.3 Inapproximability of the Minmax Value in Three Player Games

$k = 1, \dots, m$ , where

$$(R^{(k)})_{ij} = \begin{cases} 1 & \text{if } i \notin S_k^{(r)} \\ 0 & \text{if } i \in S_k^{(r)} \end{cases} \quad \text{and} \quad (C^{(k)})_{ij} = \begin{cases} 1 & \text{if } j \notin S_k^{(c)} \\ 0 & \text{if } j \in S_k^{(c)} \end{cases}$$

We claim that this game satisfies our assumptions.

For the first part, let  $S, T \subseteq [n]$  be sets of size at least  $c_2 \ln n$  such that  $d(S, T) \geq 1 - \eta$ . By choosing appropriate subsets, We may assume that, in fact,  $|S| = |T| = c_2 \ln n$ . Furthermore, by Lemma 23, with high probability

$$|S_i^r \cap S| \geq (1 - \delta)^2 c_2 \ln n \quad \text{and} \quad |S_i^c \cap T| \geq (1 - \delta)^2 c_2 \ln n.$$

Thus if players 2 and 3 play strategies  $u_S$  and  $u_T$ , respectively, playing any of the strategies  $r_k, c_k$  will give player 1 a payoff of at most

$$1 - (1 - \delta)^2 = \delta(2 - \delta) = (1 - \sqrt{1 - \eta})(1 + \sqrt{1 - \eta}) = \eta$$

while playing strategy  $b$  will give a payoff of  $1 - d(S, T) \leq \eta$ .

For the second part, we assume to the contrary that  $G$  has density  $d(S, T) < \eta$  for all sets  $S, T$  of size at least  $c_1 \ln n$ , but  $\min \max G_A \leq a$ . Let  $(\sigma_2, \sigma_3)$  be an optimal strategy profile, i.e.,

$$\max \left\{ \sigma_2^T B \sigma_3, \sigma_2^T R^{(k)} \sigma_3, \sigma_2^T C^{(k)} \sigma_3 \right\} = \min \max G_A \leq a.$$

We first show that on any support of size at most  $k_1$  each of  $\sigma_2$  and  $\sigma_3$  places probability at most  $a$ : Suppose  $S \subseteq [n]$  and  $|S| \leq k_1$  with  $\mathbb{P}_{\sigma_2}[S] = p$ . Then player 1 might increase his payoff to at least  $p$  by choosing an action  $r_k$  for which  $S_k^{(r)} \cap S = \emptyset$ . Thus  $p \leq a$ . The proof for  $\sigma_3$  is the same, replacing  $r_k$  with  $c_k$ .

We set, with foresight,

$$\begin{aligned} a &= \phi - \frac{\eta}{5}, \\ b &= 1 - \phi - \frac{\eta}{2}, \text{ and} \\ c &= 1 - \eta. \end{aligned}$$

For  $0 < \eta < \phi^2 \approx 0.146$ , these values satisfy

$$0 < a < b < c < 1 \quad b > (1 + b)a \quad \text{and} \quad (1 - a)c > b.$$

## 2 Random and Pseudorandom Structures

In fact, using  $\phi^2 = 3\phi - 1$  we get

$$\begin{aligned}
(1+b)a &= \left(2 - \phi - \frac{1}{2}\eta\right) \left(\phi - \frac{1}{5}\eta\right) \\
&= 2\phi - \phi^2 - \frac{1}{2}\phi\eta - \frac{2}{5}\eta + \frac{1}{5}\phi\eta + \frac{1}{10}\eta^2 \\
&= \left(1 - \phi - \frac{\eta}{2}\right) + \eta \left(\frac{1}{2} - \frac{1}{2}\phi - \frac{2}{5} + \frac{1}{5}\phi\right) + \frac{1}{10}\eta^2 \\
&= b + \frac{1}{10}\eta(\eta + 1 - 3\phi) \\
&= b + \frac{1}{10}\eta(\eta - \phi^2),
\end{aligned}$$

and the latter is  $< b$  if and only if  $\eta \in (0, \phi^2)$ . On the other hand,

$$\begin{aligned}
(1-a)c &= \left(1 - \phi + \frac{\eta}{5}\right) (1 - \eta) \\
&= 1 - \phi + \frac{1}{5}\eta - \eta + \phi\eta - \frac{1}{5}\eta^2 \\
&= \left(1 - \phi - \frac{1}{2}\eta\right) + \eta \left(\frac{1}{2} + \frac{1}{5} - 1 + \phi\right) - \frac{1}{5}\eta^2 \\
&= b + \eta \left(\phi - \frac{3}{10}\right) - \frac{1}{5}\eta^2 \\
&= b - \frac{1}{5}\eta \left(\eta - \left(5\phi - \frac{3}{2}\right)\right),
\end{aligned}$$

and this is  $> b$  if and only if  $\eta \in (0, 5\phi - 3/2)$ . Because  $5\phi - 3/2 \approx 0.41 > \phi^2$ , both conditions hold for  $\eta \in (0, \phi^2)$ .

We show that there exist sets  $S$  and  $T$  of size at least  $c_1 \ln n$  such that  $u_S^\top A u_T \geq 1 - c$ : Define  $T = \{i \mid \sigma_2^\top B e_i \leq b\}$ , and let  $p = \mathbb{P}_{\sigma_3}[T]$ . Then

$$a \geq \sigma_2^\top B \sigma_3 > (1-p)b,$$

and therefore  $(1-p)b < a$ . But  $b > (1+b)a$ , which implies  $p > a$ , and therefore  $|T| \geq c_1 \ln n$  by our above argument. Furthermore, by definition of  $T$  we have  $\sigma_2^\top B u_T \leq b$ .

Next, define  $S = \{i \mid e_i^\top B u_T \leq c\}$ , and let  $p = \mathbb{P}_{\sigma_2}[S]$ . Similarly to before we then have

$$b \geq \sigma_2^\top B u_T > (1-p)c$$

which means  $(1-p)c < b$ . But  $(1-a)c > b$ , which implies  $p > a$ , and again we obtain that  $|S| \geq c_1 \ln n$ . Furthermore, by definition of  $S$  and  $B = 1 - A$  we have  $u_S^\top A u_T \geq 1 - c = \eta$ .  $\square$

### 2.3.5 Derandomisation

In this section we derandomise our result in Theorem 22, at the price of turning our many-one reduction into a Turing reduction. Recall that randomness was needed by our reduction for the construction of the sets  $S_i^{(r)}$  and  $S_i^{(c)}$ . We now show how these sets can be constructed explicitly, giving a derandomised analogue of Lemma 23:

**Lemma 24.** *Let  $0 < k_1 < k_2 < n \in \mathbb{N}$ . Then there are families  $A^{(1)}, \dots, A^{(r)}$  of subsets of  $[n]$  such that*

- *there are  $r = 2^{O(k_2)} \log n$  families, and each family is of size  $s = \binom{k_2}{k_1}$ ,*
- *for every set  $M \subseteq [n]$  of size  $k_2$ , there is an index  $j \in [r]$  such that*

$$|A_i^{(j)} \cap M| = k_2 - k_1$$

*for all  $i \in [s]$  and*

- *for every set  $M \subseteq [n]$  of size  $k_1$  and every  $j \in [r]$ , there is an index  $i \in [s]$  such that*

$$A_i^{(j)} \cap M = \emptyset.$$

*These sets can be constructed in time polynomial in  $n$  and  $r$ .*

*Proof.* Recall from section 2.1.1 that a family of perfect hash functions from  $[n]$  to  $[k_2]$  is a family  $H = \{f_1, \dots, f_r\}$  such that for each  $M \subseteq [n]$  of size  $k_2$ , at least one of the  $f_j$  is injective on  $M$ . In [AYZ95], Alon et al. showed how, given  $n$  and  $k_2$ , such a family of size  $r = 2^{O(k_2)} \log n$  can be constructed deterministically in time polynomial in  $n$  and  $r$ .

Let  $s = \binom{k_2}{k_1} \leq 2^{k_2}$  and let  $M_1, \dots, M_s$  be an enumeration of the subsets of  $[k_2]$  of size  $k_1$ . Define

$$A_i^{(j)} := \{x \in [n] \mid f_j(x) \notin M_i\}.$$

These subsets meet the size restrictions claimed in the lemma and are readily seen to be constructible in time  $\text{poly}(n, r)$ .

Now, let  $M \subseteq [n]$  be of size  $k_2$ , and suppose  $f_j$  is injective on  $M$ . Then

$$A_i^{(j)} \cap M = \{x \in M \mid f_j(x) \notin M_i\},$$

and because  $f_j$  is a bijection between  $M$  and  $[k_2]$ , this set has size  $k_2 - k_1$  for all  $i \in [s]$ .

Furthermore, if  $M \subseteq [n]$  is of size  $k_1$ , then  $|f_j(M)| \leq k_1$  for all  $j \in [r]$ . Thus for each  $j$  there is an  $i$  such that

$$f_j(M) \subseteq M_i,$$

which implies  $A_i^{(j)} \cap M = \emptyset$ . □

**Corollary 25.** *If  $k_2 = O(\log n)$  then both  $r$  and  $s$  are polynomial in  $n$ , and the families of subsets can be constructed in time polynomial in  $n$ .*

## 2 Random and Pseudorandom Structures

Our derandomised reduction now looks as follows:

**Theorem 26.** *For  $0 < \eta < 0.1$  and  $0 < c_1 < c_2$  and such that*

$$\frac{c_2}{c_1} > \frac{1}{\eta},$$

*there is a polynomial-time turing reduction from Gap-DBS to Gap-Minmax with a gap  $(\eta, \phi - \eta/5)$ .*

*Proof.* The reduction works as in the randomised case, the main difference being that instead of guessing sets  $S_i^{(r)}$  and  $S_i^{(c)}$  at random, we construct (polynomially many) set families  $A^{(1)}, \dots, A^{(r)}$  using the construction in Lemma 24 with  $k_{1/2} = c_{1/2} \ln n$ . We then use each pair of such families to construct a game  $G_A^{(j_1, j_2)}$  as in the proof of Theorem 22; using the family  $A^{(j_1)}$  for the row strategies and  $A^{(j_2)}$  for the column strategies. To be precise, the 3-player game  $G_A^{(j_1, j_2)}$  looks as follows:

- Players 2 and 3 have  $n$  strategies each.
- Player 1 has  $2m + 1$  strategies denoted by

$$b^{(j_1, j_2)}, \quad r_1^{(j_1, j_2)}, \dots, r_m^{(j_1, j_2)}, \quad \text{and} \quad s_1^{(j_1, j_2)}, \dots, s_m^{(j_1, j_2)}.$$

The corresponding payoff matrices for player 1 are  $B^{(j_1, j_2)} = 1 - A$ ,  $R^{(j_1, j_2, k)}$  and  $C^{(j_1, j_2, k)}$  for  $k = 1, \dots, m$ , where

$$(R^{(j_1, j_2, k)})_{ij} = \begin{cases} 1 & \text{if } i \notin A_k^{(j_1)} \\ 0 & \text{if } i \in A_k^{(j_1)} \end{cases} \quad \text{and} \quad (C^{(j_1, j_2, k)})_{ij} = \begin{cases} 1 & \text{if } j \notin A_k^{(j_2)} \\ 0 & \text{if } j \in A_k^{(j_2)} \end{cases}.$$

We show that

- (i) if  $d(S, T) \geq 1 - \eta$  for some sets  $S, T$  of size at least  $c_2 \ln n$ , then

$$\min \max_1 G_A^{(j_1, j_2)} \leq \eta$$

for some  $j_1$  and  $j_2$ , and

- (ii) if  $d(S, T) \leq \eta$  for all sets  $S, T$  of size at least  $c_1 \ln n$ , then

$$\min \max_1 G_A^{(j_1, j_2)} \geq \phi - \eta/5$$

for all  $j_1, j_2$ .

The proof works essentially as in the randomised case: For part (i), assume  $S$  and  $T$  are sets of size at least  $c_2 \ln n$  such that  $d(S, T) \geq 1 - \eta$ . Let  $(j_1, j_2)$  be such that the set families  $A^{(j_1)}$  and  $A^{(j_2)}$  satisfy

$$|A_i^{(j_1)} \cap S| = |A_i^{(j_2)} \cap T| = k_2 - k_1 \geq (1 - \eta)k_2$$

### 2.3 Inapproximability of the Minmax Value in Three Player Games

for all  $i \in [s]$ ; such indices exist by Lemma 24. Thus if players 2 and 3 play strategies  $u_S$  and  $u_T$ , respectively, in the game  $G_A^{(j_1, j_2)}$  none of the strategies  $r_k^{(j_1, j_2)}$  and  $c_k^{(j_1, j_2)}$  will give player 1 a payoff greater than  $\eta$ . The same holds for strategy  $b^{(j_1, j_2)}$ , and therefore  $\min \max_1 G_A^{(j_1, j_2)} \leq \eta$  in this case.

For part (ii), again we assume to the contrary that  $G$  has density  $d(S, T) < \eta$  for all sets  $S, T$  of size at least  $k_1 = c_1 \ln n$ , but  $\min \max_1 G_A^{(j_1, j_2)} \leq a$  for some  $(j_1, j_2)$ . Let  $(\sigma_2, \sigma_3)$  be an optimal strategy profile, i.e.,

$$\max \left\{ \sigma_2^\top B^{(j_1, j_2)} \sigma_3, \sigma_2^\top R^{(j_1, j_2, k)} \sigma_3, \sigma_2^\top C^{(j_1, j_2, k)} \sigma_3 \right\} = \min \max_1 G_A^{(j_1, j_2)} \leq a.$$

We first show that on any support of size at most  $k_1$  each of  $\sigma_2$  and  $\sigma_3$  places probability at most  $a$ : Suppose  $S \subseteq [n]$  and  $|S| \leq k_1$  with  $\mathbb{P}_{\sigma_2}[S] = p$ . Then player 1 might increase his payoff to at least  $p$  by choosing an action  $r_k^{(j_1, j_2)}$  for which  $A_k^{(j_1)} \cap S = \emptyset$ . Thus  $p \leq a$ . The proof for  $\sigma_3$  is the same, replacing  $r_k^{(j_1, j_2)}$  with  $c_k^{(j_1, j_2)}$ .

As in the proof of Theorem 22, we set

$$\begin{aligned} a &= \phi - \frac{\eta}{5}, \\ b &= 1 - \phi - \frac{3\eta}{5}, \text{ and} \\ c &= 1 - \eta \end{aligned}$$

and recall that these satisfy

$$0 < a < b < c < 1 \quad b > (1 + b)a \quad \text{and} \quad (1 - a)c > b$$

for  $0 < \eta < \phi^2$ . We show that there exist sets  $S$  and  $T$  of size at least  $c_1 \ln n$  such that  $u_S^\top A u_T \geq 1 - c$ : Define  $T = \{i \mid \sigma_2^\top B^{(j_1, j_2)} e_i \leq b\}$ , and let  $p = \mathbb{P}_{\sigma_3}[T]$ . Then

$$a \geq \sigma_2^\top B^{(j_1, j_2)} \sigma_3 > (1 - p)b,$$

and therefore  $(1 - p)b < a$ . But  $b > (1 + b)a$ , which implies  $p > a$ , and therefore  $|T| \geq c_1 \ln n$  by our above argument. Furthermore, by definition of  $T$  we have  $\sigma_2^\top B^{(j_1, j_2)} u_T \leq b$ .

Next, define  $S = \{i \mid e_i^\top B^{(j_1, j_2)} u_T \leq c\}$ , and let  $p = \mathbb{P}_{\sigma_2}[S]$ . Similarly to before we then have

$$b \geq \sigma_2^\top B^{(j_1, j_2)} u_T > (1 - p)c$$

which means  $(1 - p)c < b$ . But  $(1 - a)c > b$ , which implies  $p > a$ , and again we obtain that  $|S| \geq c_1 \ln n$ . Furthermore, by definition of  $S$  and  $B^{(j_1, j_2)} = 1 - A$  we have  $u_S^\top A u_T \geq 1 - c = \eta$ .  $\square$

## Notes

The results in section 2.2 were obtained together with Martin Grohe and Magdalena Grüber and published in [EGG08]. In [Mar10], Dániel Marx improved upon that result

## 2 Random and Pseudorandom Structures

by showing that, unless  $W[P] = FPT$ , there is no fpt-cost approximation algorithm for  $\text{MIN-WSAT}(\text{CIRC}^+)$  for *any* computable ratio  $\rho$ . Furthermore, Marx showed that if  $W[1] \neq FPT$ , then the dual problem  $\text{MAX-WSAT}(\text{CIRC}^-)$  cannot be fpt-cost approximated either, for any computable ratio  $\rho$ . In fact, he proposes a new problem which is in  $W[2]$  and which he shows to be non-approximable. His proof is fundamentally different from ours.

The results in this section 2.3 are based on joint work with Kristoffer Arnsfelt Hansen and Elad Verbin at Aarhus University. They have been submitted and are presently under review [EHV11].



## 3 Randomised Logics

In this and the next chapter we will introduce randomised logics and study their expressive power compared to their non-randomised counterparts. Our goal in studying randomised logics is to apply tools from descriptive complexity theory to the study of randomised complexity classes. To this end, we first introduce a general way of introducing randomness to logics, similar to the BP and R operators of complexity theory (cf. Section 1.4). Thus, to any logic  $L$  we associate randomised variants BPL and RL. We then show for certain choices of  $L$  that the resulting logics indeed capture well-known randomised complexity classes.

That descriptive complexity indeed offers new insights into randomised complexity classes is witnessed by the results in Section 3.4: We show that there are queries definable in randomised first-order logic but not in various non-randomised logics. This, in particular, implies that a certain strongly uniform variant of randomised  $AC^0$  can provably *not* be derandomised. Since there are uniform variants of  $AC^0$  which can be derandomised, this opens the question of where the boundary between these two cases lies. The major open question here is whether dlogtime-uniform  $BPAC^0$  can be derandomised or not.

### 3.1 Randomised logics

Throughout this section, let  $\tau$  and  $\rho$  be disjoint vocabularies. Relations over  $\rho$  will be “random”, and we will reserve the letter  $R$  for relation symbols from  $\rho$ . We are interested in *random*  $(\tau \cup \rho)$ -*expansions* of  $\tau$ -structures. For a  $\tau$ -structure  $A$ , by  $\mathcal{X}(A, \rho)$  we denote the class of all  $(\tau \cup \rho)$ -expansions of  $A$ . We view  $\mathcal{X}(A, \rho)$  as a probability space with the uniform distribution. Note that we can “construct” a random  $X \in \mathcal{X}(A, \rho)$  by deciding independently for all  $k$ -ary  $R \in \rho$  and all tuples  $\vec{a} \in V(A)^k$  with probability  $1/2$  whether  $\vec{a} \in R(X)$ . Hence if  $\rho = \{R_1, \dots, R_k\}$ , where  $R_i$  is  $r_i$ -ary, then a random  $X \in \mathcal{X}(A, \rho)$  can be described by a random bitstring of length  $\sum_{i=1}^k n^{r_i}$ , where  $n := |V(A)|$ . We are mainly interested in the probabilities

$$\mathbb{P}_{X \in \mathcal{X}(A, \rho)} (X \models \phi)$$

that a random  $(\tau \cup \rho)$ -expansion of a  $\tau$ -structure  $A$  satisfies a sentence  $\phi$  of vocabulary  $\tau \cup \rho$  of some logic. Notice that, if  $A$  has non-trivial automorphisms,  $\mathcal{X}(A, \rho)$  will contain at least two distinct but isomorphic structures, and we count these individually. Equivalently, we may assume that  $A$  is actually ordered, though the ordering is not necessarily accessible to the logic.

### 3 Randomised Logics

**Definition 27.** Let  $\mathbf{L}$  be a logic and  $0 \leq \alpha \leq \beta \leq 1$ .

1. A formula  $\phi \in \mathbf{L}[\tau \cup \rho]$  that defines a  $k$ -ary query has an  $(\alpha, \beta]$ -gap if for all  $\tau$ -structures  $A$  and all  $\vec{a} \in V(A)^k$  it holds that

$$\mathbb{P}_{X \in \mathcal{X}(A, \rho)}(X \models \phi[\vec{a}]) \leq \alpha \quad \text{or} \quad \mathbb{P}_{X \in \mathcal{X}(A, \rho)}(X \models \phi[\vec{a}]) > \beta.$$

2. The logic  $\mathbf{P}_{(\alpha, \beta]} \mathbf{L}$  is defined as follows: For each vocabulary  $\tau$ ,

$$\mathbf{P}_{(\alpha, \beta]} \mathbf{L}[\tau] := \bigcup_{\rho} \{ \phi \in \mathbf{L}[\tau \cup \rho] \mid \phi \text{ has an } (\alpha, \beta]\text{-gap} \},$$

where the union ranges over all vocabularies  $\rho$  disjoint from  $\tau$ . To define the semantics, let  $\phi \in \mathbf{P}_{(\alpha, \beta]} \mathbf{L}[\tau]$ . Let  $k, \rho$  such that  $\phi \in \mathbf{L}[\tau \cup \rho]$  and  $\phi$  is  $k$ -ary. Then for all  $\tau$ -structures  $A$ ,

$$\mathcal{Q}_{\phi}^{\mathbf{P}_{(\alpha, \beta]} \mathbf{L}}(A) := \{ \vec{a} \in V(A)^k \mid \mathbb{P}_{X \in \mathcal{X}(A, \rho)}(X \models_{\mathbf{L}} \phi[\vec{a}]) > \beta \}.$$

It is easy to see that for every logic  $\mathbf{L}$  and all  $\alpha, \beta$  with  $0 \leq \alpha \leq \beta \leq 1$  the logic  $\mathbf{P}_{(\alpha, \beta]} \mathbf{L}$  satisfies conditions (i) to (iii) of section 1.3.2, so it again forms a well-defined logic.

We let

$$\mathbf{PL} := \mathbf{P}_{(1/2, 1/2]} \mathbf{L} \quad \text{and} \quad \mathbf{RL} := \mathbf{P}_{(0, 2/3]} \mathbf{L} \quad \text{and} \quad \mathbf{BPL} := \mathbf{P}_{(1/3, 2/3]} \mathbf{L}.$$

We can also define a logic  $\mathbf{P}_{[\alpha, \beta]} \mathbf{L}$  and let  $\mathbf{co-RL} := \mathbf{P}_{[1/3, 1]} \mathbf{L}$ . The following lemma shows that for reasonable  $\mathbf{L}$  the strength of the logic  $\mathbf{P}_{(\alpha, \beta]} \mathbf{L}$  does not depend on the exact choice of the parameters  $\alpha, \beta$ . This justifies the arbitrary choice of the constants  $1/3, 2/3$  in the definitions of  $\mathbf{RL}$  and  $\mathbf{BPL}$ .

**Lemma 28.** *Let  $\mathbf{L}$  be a logic that is closed under conjunctions and disjunctions. Then for all  $\alpha, \beta$  with  $0 < \alpha < \beta < 1$  it holds that  $\mathbf{P}_{(0, \beta]} \mathbf{L} \equiv \mathbf{RL}$  and  $\mathbf{P}_{(\alpha, \beta]} \mathbf{L} \equiv \mathbf{BPL}$ .*

*Proof.* Let  $\tau$  and  $\rho = \{R_1, \dots, R_k\}$  be disjoint relational vocabularies and let  $\varphi \in \mathbf{L}[\tau \cup \rho]$ . For any  $n \geq 1$  we define a new vocabulary

$$\rho^{(n)} := \{R_j^{(i)} \mid 1 \leq i \leq n, 1 \leq j \leq k\},$$

where the arity of  $R_j^{(i)}$  is that of  $R_j \in \rho$ . Using the renaming property with the renaming

$$r^{(i)} : (\tau \cup \rho) \rightarrow (\tau \cup \rho^{(n)})$$

that leaves  $\tau$  fixed and maps  $R_j \in \rho$  to  $R_j^{(i)}$  we get sentences  $\varphi^{(i)}$ , which are the sentence  $\varphi$  with every occurrence of  $R_j$  replaced by  $R_j^{(i)}$ . Since  $\mathbf{L}$  is closed under conjunctions and

### 3.2 Previous Work on Randomised Logics

disjunctions, for every  $0 < l \leq n$  there is an  $\mathbb{L}[\tau \cup \rho^{(n)}]$ -sentence

$$\varphi^{(n,l)} := \bigvee_{\substack{I \subseteq [n] \\ |I|=l}} \bigwedge_{i \in I} \varphi^{(i)}$$

which is satisfied iff at least  $l$  of the  $\varphi^{(i)}$  are satisfied. Notice that the  $\varphi^{(i)}$  use distinct random relations, so they are satisfied independently of each other.

Clearly, if  $\mathbb{P}(X \models \varphi) = 0$  then also  $\mathbb{P}(X \models \varphi^{(n,l)}) = 0$ , because we assumed  $l \geq 1$ . On the other hand, if  $\mathbb{P}(X \models \varphi) > \beta$  for some  $\beta \in (0, 1)$ , then

$$\mathbb{P}(X \models \varphi^{(n,1)}) = 1 - (1 - \mathbb{P}(X \models \varphi))^n \quad (3.1)$$

$$> 1 - (1 - \beta)^n, \quad (3.2)$$

and this bound can be made arbitrarily close to 1 by choosing  $n$  sufficiently large. This proves the claim about RL.

For BPL, notice that if  $\varphi$  has an  $(\alpha, \beta]$ -gap for some any  $0 < \alpha < \beta < 1$ , then for any  $0 < \alpha' < \beta' < 1$  there is an  $n \in \mathbb{N}$  such that

$$\varphi^{(n, \lceil \frac{\beta - \alpha}{2} \rceil)}$$

has an  $(\alpha', \beta']$ -gap. In fact, the Chernoff bound (cf. Theorem 1) gives very sharp estimates on  $n$  in terms of  $\alpha$ ,  $\beta$ ,  $\alpha'$  and  $\beta'$ , though we only need the mere existence of such an  $n$  here.  $\square$

### 3.2 Previous Work on Randomised Logics

While there has been extensive research on randomised computation, both in the area of complexity theory and in algorithm design, randomised logics have received rather less interest. The main point which distinguishes our work from previous research is that we consider *partly* random structures, i.e., structures which come with some pre-defined relations as well as random ones.

Arguably the best-known results on the behaviour of logical sentences on *purely* random structures are the classical 0-1-laws, discovered independently by Fagin [Fag76] and Glebskiĭ et al. [GKLT69]. These state that for any sentence  $\varphi$  in first-order logic (or, in fact,  $\mathbb{L}_{\infty\omega}^\omega$ ; cf. [KV92]), the probability that  $\varphi$  is satisfied in a random structure of size  $n$  tends either to 0 or to 1 as  $n$  goes to infinity. In our notation, this reads

$$\lim_{n \rightarrow \infty} \mathbb{P}([n] \models \varphi) \in \{0, 1\},$$

and in particular the limit exists. Here,  $[n]$  is a set of  $n$  elements with no further structure on it. Note that this holds only for purely relational vocabularies. A consequence of this is that, on structures over the empty vocabulary,  $\mathbb{L}_{\infty\omega}^\omega$  gains no expressive power by randomisation, cf. Observation 42 below.

Extensions to this result have mostly dealt with probability distributions other than

the uniform one. For example, Spencer [Spe01] proved that 0-1-laws hold in random graphs with edge probability  $n^{-\alpha}$  for every irrational  $\alpha$ . That is, the vocabulary of random relations contains just a single binary relation symbol  $E$ , and the structures are drawn from all structures in which  $E$  is symmetric and irreflexive according to the Erdős-Rényi-graph model where each edge is present independently with probability  $n^{-\alpha}$  (cf. [Bol01]).

The only significant work on partially random structures which we are aware of is by Shelah [She96] and Boppana and Spencer [BS95], who prove what they call *smoothness law* or *very weak 0-1-law* for first-order logic. They consider structures whose non-random part is fixed to be a linear order. They also restrict the random part to an Erdős-Rényi random graph  $G_{n,1/2}$ , i.e., they only consider a single binary relation which is drawn uniformly from among all symmetric and irreflexive relations, but unlike the restriction on the non-random part this one is immaterial and can easily be removed from their arguments. They show that there is not even a convergence law for first-order logic on ordered random structures<sup>1</sup>, i.e., there are FO-sentences  $\varphi$  for which

$$\lim_{n \rightarrow \infty} \Pr(\mathcal{O}_n \models \varphi)$$

does not exist; recall that  $\mathcal{O}_n = \mathcal{N}_n|_{\{\leq\}}$  denotes a linear order of size  $n$ . On the other hand, using techniques similar to the ones we apply in Section 4.1 they obtain the following bound for the oscillation of this probability: For large enough  $n$ ,

$$|\Pr(\mathcal{O}_n \models \varphi) - \Pr(\mathcal{O}_{n+1} \models \varphi)| \leq \frac{\log^{O(1)} n}{n},$$

which implies that, on the class of linear orders, BPFO can be derandomised (cf. Theorem 49).

Finally, in [Mül08] Moritz Müller introduced logics with random quantifiers, i.e., quantifiers which quantify over universe elements which are drawn uniformly at random. In our logics, we use random relations rather than random universe elements, and we do not allow quantification but instead force a kind of prenex normal form by designating certain relation symbols to be interpreted randomly. This is because, in order to capture meaningful randomised complexity classes, we usually insist on an error gap, which is harder to handle if one allows quantifiers.

### 3.3 Capturing Results

Our motivation for studying randomised logics comes from computational complexity theory. A natural question to ask is therefore whether the logics we define do indeed capture randomised complexity classes of interest. In this section we give examples of capturing results which show that this is indeed the case, and therefore we can apply techniques from finite model-theory to the study of these randomised complexity classes. In particular, we show that

---

<sup>1</sup>In this case the ordering is accessible to the logic.

- BPFO captures  $\text{BPAC}^0$  on ordered structures, structures with addition and structures with addition and multiplication. This extends the corresponding capturing results for FO and  $\text{AC}^0$  (cf. theorem 11 and theorem 12).
- BPIFP+C (i.e., randomised inflationary fixed-point logic with counting) captures BPP on *all* structures, in particular also on unordered ones. This extends Immerman and Vardi's theorem that IFP captures PTIME on ordered structures (cf. theorem 9). Note that, on ordered structures,  $\text{IFP}+\text{C} \equiv \text{IFP}$ , because counting may be realised using fixed-point operators on these structures. However, Cai et al. [CFI92] showed that  $\text{IFP}+\text{C}$  does *not* capture PTIME on all structures (they even show this for the stronger logic  $\text{C}_{\infty\omega}^\omega$ ), whereas in the randomised case we get a logic capturing BPP on *all* structures. Note however that the logic BPIFP+C has an undecidable syntax, so even if one assumes that  $\text{BPP} = \text{PTIME}$  this result sheds no light on the question of whether or not there is a logic capturing PTIME in the sense of [Gro08].

These are by no means the only cases of randomised logics capturing randomised complexity classes, but we will not dwell on other results of this type here. For example, it is easy to see that randomised existential second-order logic  $\text{R}\Sigma_1$  captures the Arthur-Merlin complexity class AM. There are some caveats when translating classical capturing results to randomised ones, though, in particular with classes like BPL (randomised LOGSPACE), which are usually defined by allowing only read-once access to the random bits.

### 3.3.1 BPFO Captures $\text{BPAC}^0$ on Ordered Structures

Recall Barrington et al.'s result (Theorem 11) stating that, on structures with addition and multiplication (or, equivalently, with a bit predicate), first-order logic captures dlogtime-uniform  $\text{AC}^0$ . This, as well as Behle and Lange's extension (Theorem 12) to stricter uniformity conditions, carries over to the randomised world. To be precise:

**Theorem 29.** *Let  $\tau$  be either  $\{\leq\}$ ,  $\{+\}$  or  $\{\text{Bit}\}$ . Then  $\text{BPFO}[\tau]$  captures  $\text{FO}[\tau]$ -uniform  $\text{BPAC}^0$ . In particular,  $\text{BPFO}[\text{Bit}]$  captures dlogtime-uniform  $\text{BPAC}^0$ .*

*Proof.* The crucial observation here is that a linear order, addition relation and multiplication relation may be defined on *tuples* of fixed length in first-order logic from the corresponding relations. E.g., the following formula defines the lexicographic ordering on pairs from a given ordering  $\leq$ :

$$\varphi_{\leq}(x, y, u, v) := (x \leq u) \vee (x \dot{=} u \wedge y \leq v)$$

The existence of similar formulas of arbitrary width for  $+$  and  $\text{Bit}$  is shown in [Sch05].

Let  $(C_n)_{n \geq 1}$  be an  $\text{BPAC}^0$  circuit family satisfying one of the uniformity conditions above. We may assume that the circuit for inputs of length  $n$  has exactly  $n^c$  inputs for some constant  $c \in \mathbb{N}$  independent of  $n$ , of which the last  $n^c - n$  are random ones.

### 3 Randomised Logics

By Theorems 11 and 12, there is an FO-sentence  $\varphi$  such that for every  $n \in \mathbb{N}$  and  $x \in \{0, 1\}^n$ ,

$$C_{|x|^c} \text{ accepts } (x, r) \quad \text{iff} \quad w_{(x,r)} \models \varphi,$$

where  $r \in \{0, 1\}^{n^c-n}$  is a string of random bits and  $w_{(x,r)}$  is the word model for the concatenated string  $xr$  with the appropriate built-in relations.

We interpret the structure  $W_{(x,r)}$  within the structure  $W_x$  by using a  $c$ -ary random relation. To this end, we replace every quantifier  $\forall y$  with a quantifier block  $\forall \vec{y}$ , where  $\vec{y}$  is a block of  $c$  variables, and similarly for existential quantifiers. We replace the built-in relations with formulas for their  $c$ -ary counterparts. Finally, assume  $P$  is the unary relation symbol in the vocabulary of  $W_x$  such that  $i \in P^{W_x}$  iff  $x_i = 1$ . We replace  $Py$  with  $\varphi_P(\vec{y})$ , where

$$\varphi_P(\vec{y}) := (y_2 \dot{=} 0 \wedge y_3 \dot{=} 0 \wedge \dots \wedge y_c \dot{=} 0 \wedge Py_1) \vee (\neg(y_2 \dot{=} 0 \wedge y_3 \dot{=} 0 \wedge \dots \wedge y_c \dot{=} 0) \wedge R\vec{y})$$

The resulting sentence  $\psi$  satisfies

$$\mathbb{P}_r(C_{|x|^c} \text{ accepts } (x, r)) = \mathbb{P}(W_x \models \psi),$$

as was required. □

#### 3.3.2 A Logic Capturing BPP

In this section, we prove that the logic BPIFP+C captures the complexity class BPP, even on unordered structures. Technically, the results of this section are closely related to results in [HKL96].

Counting logics like FO+C and IFP+C are usually defined via two-sorted structures, which are equipped with an initial segment of the natural numbers of appropriate length. The expressive power of the resulting logic turns out to be rather robust under changes in the exact definition, see [Ott96] for a detailed survey of this. However, we will only need the limited counting ability provided by the *Rescher quantifier*, which goes back to a unary majority quantifier defined in [Res62], see [Ott96].

We let  $\text{FO}(\mathcal{J})$  be the logic obtained from first-order logic by adjoining a generalised quantifier  $\mathcal{J}$ , the *Rescher quantifier*. For any two formulas  $\varphi_1(\vec{x})$  and  $\varphi_2(\vec{x})$ , where  $\vec{x}$  is a  $k$ -tuple of variables, we form a new formula

$$\mathcal{J}\vec{x}.\varphi_1(\vec{x})\varphi_2(\vec{x}).$$

Its semantics is defined by

$$A \models \mathcal{J}\vec{x}.\varphi_1(\vec{x})\varphi_2(\vec{x}) \quad \text{iff} \quad \left| \{ \vec{a} \in V(A)^k \mid A \models \varphi_1[\vec{a}] \} \right| \leq \left| \{ \vec{a} \in V(A)^k \mid A \models \varphi_2[\vec{a}] \} \right|. \quad (3.3)$$

The logic IFP( $\mathcal{J}$ ) is defined similarly.

**Lemma 30.** *Let  $R$  be a 6-ary relation symbol. There is a formula*

$$\phi_{\leq}(x, y) \in \text{FO}(\mathcal{J})[\{R\}]$$

*such that*

$$\lim_{n \rightarrow \infty} \mathbb{P}_{A \in \mathcal{X}([n], \{R\})} \left( \{(a, b) \mid A \models \phi_{\leq}[a, b]\} \text{ is a linear order of } V(A) \right) = 1.$$

(Here,  $[n]$  is the  $\emptyset$ -structure with universe  $\{1, \dots, n\}$ . Thus  $\mathcal{X}([n], \{R\})$  just denotes the set of all  $\{R\}$ -structures with universe  $\{1, \dots, n\}$ .)

*Proof.* We let

$$\phi_{\leq}(x, y) := \mathcal{J}x_1 \dots x_5. Rxx_1 \dots x_5 Ryx_1 \dots x_5.$$

To see that  $\phi_{\leq}(x, y)$  defines an order with high probability, let  $A$  be a structure with universe  $V(A) = \{1, \dots, n\}$ . For each  $a \in V(A)$ , let

$$X_a := \left| \{\vec{a} \in V(A)^5 \mid A \models Ra\vec{a}\} \right|.$$

Then  $A \models \phi_{\leq}(a, b)$  iff  $X_a \leq X_b$ , and  $\phi_{\leq}$  linearly orders  $A$  iff the  $X_a$  are pairwise distinct. But for  $a \neq b \in V(A)$ , the random variables  $X_a$  and  $X_b$  are independent and each is binomially distributed with parameters  $p = 1/2$  and  $m = n^5$ , and thus

$$\begin{aligned} \mathbb{P}(X_a = X_b) &= \sum_{k=0}^m \left( \frac{1}{2^m} \binom{m}{k} \right)^2 \\ &= \frac{1}{2^{2m}} \sum_{k=0}^m \binom{m}{k}^2 \\ &= \frac{1}{2^{2m}} \sum_{k=0}^m \binom{m}{k} \binom{m}{m-k} \\ &= \frac{1}{2^{2m}} \binom{2m}{m} \\ &= \Theta\left(\frac{1}{\sqrt{m}}\right), \end{aligned}$$

where the final approximation can be found, for example, in [Fel57]. Taking a union bound over all  $\binom{n}{2} = \Theta(m^{2/5})$  pairs  $a \neq b$  gives the desired result.  $\square$

**Remark 31.** While using a 6-ary relation makes the above analysis of the success probability particularly simple, in IFP it is also possible to define an order with high probability using a binary random relation and Rescher quantifier [BES80] or a binary random relation and an even quantifier [HKL96].

**Theorem 32.** *The logic  $\text{BPIFP}(\mathcal{J})$  captures BPP.*

### 3 Randomised Logics

*Proof.*  $\text{BPIFP}(\mathcal{J})$  is contained in BPP, because a randomised polynomial time algorithm can interpret the random relations by using its random bits.

For the other direction, let  $\mathcal{Q}$  be a Boolean query in BPP. This means that there is a randomised polynomial time algorithm  $M$  that decides the query  $\mathcal{Q}_{\leq}$  of ordered expansions of structures in  $\mathcal{Q}$ . We may view the (polynomially many) random bits used by  $M$  as part of the input. Then it follows from the Immerman-Vardi Theorem that there is a BPIFP-sentence  $\psi_M$  defining  $\mathcal{Q}_{\leq}$ . Note that, by the definition of  $\mathcal{Q}_{\leq}$ , this sentence is order-invariant. We replace every occurrence of  $\leq$  in  $\psi_M$  by the formula  $\varphi_{\leq}(x, y)$  of Lemma 30, which with high probability defines a linear order on the universe.  $\square$

It is easy to see that  $\text{BPIFP}+\text{C}$  is also contained in BPP and that  $\text{IFP}(\mathcal{J}) \preceq \text{IFP}+\text{C}$ . Thus we get the following corollary.

**Corollary 33.**  $\text{BPIFP}+\text{C} = \text{BPIFP}(\mathcal{J})$ , and both capture BPP.

**Remark 34.** Lemma 30 also implies that  $\text{BPL}_{\infty\omega}^{\omega}(\mathcal{J}) \equiv \text{BPC}_{\infty\omega}^{\omega}$ , because, in the presence of an ordering, a quantifier of the form  $\exists^{\geq n}x \varphi$  may be spelled out as

$$\bigvee_{\substack{S \subseteq \mathbb{N} \\ |S|=n}} \bigwedge_{i \in S} \exists x (\varphi_{i\text{-th}}(x) \wedge \varphi(x)),$$

where  $\varphi_{i\text{-th}}(x)$  holds iff  $x$  is the  $i$ -th element in the linear order; this can be done using three variables by

$$\begin{aligned} \varphi_{1\text{-th}}(x) &:= \forall y \, x \leq y \\ \varphi_{(n+1)\text{-th}}(x) &:= \exists y \forall z \, (\varphi_{n\text{-th}}(y) \wedge \neg(x \dot{=} y) \wedge y \leq x \wedge \\ &\quad ((y \leq z \wedge z \leq x) \rightarrow (y \dot{=} z \vee y \dot{=} z))). \end{aligned}$$

In fact, because of these formulas, *any* query is definable in  $\text{L}_{\infty\omega}^{\omega}$  on ordered structures, as well as on  $\text{BPC}_{\infty\omega}^{\omega}$ .

## 3.4 Separation Results

While for complexity classes such as BPP and dlogtime-uniform  $\text{BPAC}^0$  it is generally believed that these can be derandomised, we show the following results:

- RFO is not contained in  $\text{C}_{\infty\omega}^{\omega}$
- BPFO is not contained in MSO on ordered structures
- RFO is stronger than FO on additive structures

A fortiori, the first and the third result also hold with BPFO instead of RFO, and the constructions used in their proofs also admit co-RFO-definitions.



It turns out that we need three rather different queries to get these separation results. For the first two queries this is immediate by the fact that *any* query on ordered structures is definable in  $C_{\infty\omega}^\omega$ . The third query (on additive structures) is readily seen to be definable in MSO and will be complemented in chapter 4 by the following result:

- Any BPFO-definable query on additive structures can be defined in MSO.

The fact that RFO is stronger than FO on additive structures has direct consequences in classical complexity theory: By Behle and Lange’s result (cf. 12), it implies that  $FO[+]$ -uniform  $BPAC^0$  can *not* be derandomised. While  $FO[+]$ -uniformity is a very strong uniformity condition that has not received much attention so far, our result raises the question of where the boundary between derandomisable and non-derandomisable uniform variants of  $BPAC^0$  lies. Under the (rather weak) uniformity condition of EXPTIME-uniformity,  $BPAC^0$  can be derandomised, because in EXPTIME we can do a brute-force search to find a random string which gives the correct answer for all inputs of a specific size, and hard-code this string into the circuit. Dlogtime-uniformity (or, equivalently,  $FO[+, \times]$ -uniformity) falls between these two uniformity conditions, and the question of whether or not dlogtime-uniform  $BPAC^0$  can be derandomised has been studied, among others, by Viola [Vio04], who gave a conditional derandomisation. However, while there are explicit winning strategies for Ehrenfeucht-Fraïssé games on additive structures (which we use in the proof of theorem 39), finding such strategies for structures with addition and multiplication seems much more difficult.

### 3.4.1 RFO is Not Contained in $C_{\infty\omega}^\omega$

Recall from Section 1.3 that formulas of the logic  $C_{\infty\omega}^\omega$  may contain arbitrary (not necessarily finite) conjunctions and disjunctions, but only finitely many variables, and counting quantifiers of the form  $\exists^{\geq n} x \varphi$  (“there exist at least  $n$   $x$  such that  $\varphi$ ”). For example, the class of finite structures of even cardinality can be defined in this logic by the sentence

$$\bigvee_{k \geq 0} \left( \exists^{\geq 2k} x. x \dot{=} x \right) \wedge \neg \left( \exists^{\geq 2k+1} x. x \dot{=} x \right).$$

In light of Observation 42, which stated that on empty vocabularies BPFO is no stronger than FO, the above example gives a query (i.e., evenness) which is definable in  $FO+C \preceq C_{\infty\omega}^\omega$  but not in BPFO.

Interest in the infinitary counting logic  $C_{\infty\omega}^\omega$  is mainly motivated by the fact that, while it is easier to reason about than the more interesting (and strictly weaker) fixed-point logic with counting  $IFP+C$ , it still allows for strong non-definability results, which a fortiori also apply to any weaker logic. The most important result in this context is Cai, Fürer and Immerman’s proof that  $C_{\infty\omega}^\omega$  does not capture PTIME on the class of all structures [CFI92]. Their proof exhibits a certain query on graphs which is decidable in PTIME but not definable in  $C_{\infty\omega}^\omega$ . By appropriately modifying their query, we prove the following result:

**Theorem 35.** *There is a class  $\mathcal{TCFI}$  of structures that is definable in RFO and co-RFO, but not in  $C_{\infty\omega}^\omega$ .*

Our modification of Cai et al.’s query is somewhat reminiscent to proofs by Dawar, Hella, and Kolaitis [DHK95] for results on implicit definability in first-order logic. Just like in Cai, Fürer and Immerman’s original proof, the reason why  $C_{\infty\omega}^\omega$  can not define our query  $\mathcal{TCFI}$  is its inability to choose one out of a pair of two elements. Using a random binary relation this can – with high probability – be done in FO.

We first review the construction of [CFI92] and then show how to modify it to suit our needs. Given a graph  $G = (V, E)$ , Cai et al. construct a new graph  $G'$ , replacing all vertices and edges of  $G$  with certain gadgets. We shall call graphs  $G'$  resulting in this fashion *CFI-graphs*, and will from now on restrict ourselves to connected 3-regular graphs  $G$  and CFI-graphs resulting from these.

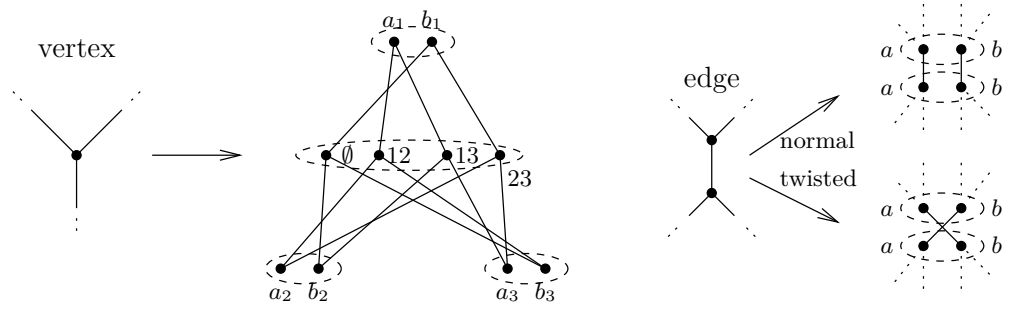


Figure 3.1: The gadgets for CFI-graphs. Dashed ellipses indicate groups of equivalent vertices. Vertex labels are not part of the actual structure.

The construction is as follows: For each vertex in  $G$ , we place a copy of the gadget shown on the left of Figure 3.1 in  $G'$ . It has a group of four nodes (henceforth called *centre nodes*) plus three pairs of nodes, which are to be thought of as ends of the three edges incident with that node. For the time being, we think of the pairs as ordered from 1 to 3 and distinguish between the two nodes in each pair, say one of them is the  $a$ -node, the other one being the  $b$  node. Each of the four centre nodes is connected to one node from each pair, and each of them to an even number of  $a$ ’s. To illustrate this, the centre nodes are labelled with the even subsets of  $\{1, 2, 3\}$ .

For each edge in  $G$ , we connect the  $a$ - and  $b$ -nodes in the corresponding pairs as shown on the right of Figure 3.1. We say an edge is “twisted” if the  $a$ -node of one pair is connected to the  $b$ -node of the other and vice versa. This completes our construction of  $G'$ . For definiteness, when we speak of an *edge group* we mean an equivalence class of size two, and by a *centre group* we mean one of size four. An *edget* is a pair of edge groups which form an edge gadget as on the right of Figure 3.1. Figure 3.2 shows the result of applying this construction to a small subgraph (a vertex with its three neighbours).

Without the  $a$ - and  $b$ -labels, we cannot decide which of the edges have been twisted. In fact there are only two isomorphism classes of CFI-graphs derived from  $G$ , namely

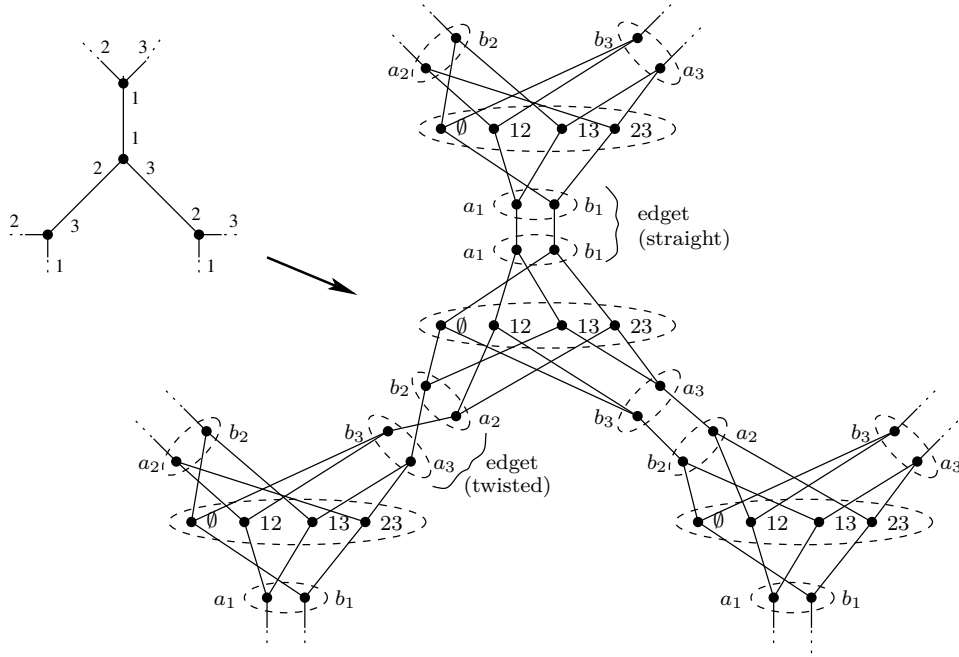


Figure 3.2: The CFI-graph construction for a part of a graph. Edge and nodes labels are not part of the actual graph.

those with an even number of edges twisted and those with an odd number (we call the latter ones *twisted* CFI-graphs). This relies on the fact that isomorphisms of the gadget on the left of Figure 3.1 are exactly those permutations swapping an even number of  $a$ 's and  $b$ 's. Since we assume  $G$  to be connected, we can twist edges along a path between two nodes adjacent to twisted edges, reducing the number of twisted edges by two.

Now, for every  $C_{\infty\omega}^\omega$ -sentence  $\varphi$ , if the original graph  $G$  is complicated enough, the two isomorphism classes can not be told apart by  $\varphi$  [CFI92]. In PTIME, on the other hand, twisted CFI-graphs can easily be recognised: Choose exactly one node from each edge group and label this one  $a$  and the other one  $b$ . A centre node is connected to an even number of  $a$ 's if and only if all four nodes in its centre group are. In this case we call the centre group even, otherwise we call it odd. Then a CFI-graph is twisted if and only if

$$(\text{no. of odd centre groups} + \text{no. of twisted edgets}) \text{ is odd.}$$

We aim for a (co-)RFO-sentence which defines exactly the twisted connected 3-regular CFI-graphs. In view of the above PTIME-algorithm, we are done if we can

- express connectedness of the graph,
- count modulo two and
- choose one representative from each centre group, edge group and edget.

### 3 Randomised Logics

For counting modulo two and to get representatives for centre groups and edgets, we augment the structures with a Boolean algebra in the following way: Let  $\tau$  be the vocabulary  $\{E, \sim, <, \sqsubseteq, P, O\}$ , with unary  $P$  and  $O$ , and binary  $E, \sim, <$  and  $\sqsubseteq$ . Let  $\mathcal{CFI}$  be the class of structures  $A$  such that

- $E(A)$  is the edge set of a 3-regular, connected CFI-graph on  $V(A) \setminus P(A)$ ,
- $(P(A), \sqsubseteq(A))$  is a Boolean algebra  $\mathfrak{B}$ , and  $O$  is true exactly for its members of even cardinality,
- $<(A)$  is a linear order on the set of atoms of  $\mathfrak{B}$  (and no other element of  $A$  is  $<$ -related to any other),
- $\sim(A)$  is an equivalence relation, where each equivalence class
  - either contains one atom and the nodes of one edget
  - or consists of a single non-atom of  $\mathfrak{B}$ .

**Theorem 36.** *The class  $\mathcal{CFI}$  is definable in FO. The subclass  $\mathcal{TCFI}$  of twisted CFI-graphs is definable in RFO and co-RFO (and therefore, in particular, in BPFO) but not in  $\mathcal{C}_{\infty\omega}^\omega$ .*

*Proof.* That  $\mathcal{CFI}$  is definable is easy to establish, the only subtlety being that  $\mathfrak{B}$  allows us to quantify over sets of centre groups, which makes connectedness expressible.

The proof that  $\mathcal{TCFI}$  is not definable in  $\mathcal{C}_{\infty\omega}^\omega$  is the same as in [CFI92]; it is unaffected by the additional structure. Note that because the atoms are ordered, the Boolean algebra is rigid, i.e., it has no non-trivial automorphism, therefore the isomorphism group of a CFI-graph is not changed by adding the Boolean algebra.

It remains to show that twistedness can be defined in BPFO. We pick one vertex from each edge group by viewing a random binary relation  $R$  as assigning an  $m$ -bit number to each vertex, where  $m$  is the number of atoms in the Boolean algebra. From each pair, we choose the vertex with the smaller number, expressed by

$$\xi(x) := \exists y \left( x \sim y \wedge \exists z (\alpha(z) \wedge \neg R x z \wedge R y z \wedge \forall w (w < z \rightarrow (R x w \leftrightarrow R y w))) \right),$$

where  $\alpha(x)$  is an FO-formula satisfied exactly by the atoms of the Boolean algebra. It is easy to see that if the random relation  $R$  assigns a different set of atoms to the two vertices in each edge group, then  $\xi$  succeeds in picking exactly one vertex from each edge group, and twistedness can then be checked by looking at the  $O$ -predicate of the element of  $\mathfrak{B}$  which contains exactly the atoms equivalent to twisted centre groups or twisted edgets.

To prove that the resulting formula has a large probability gap, we need to establish a high probability of success only for structures in the class  $\mathcal{CFI}$ , because this class is FO-definable. But in such structures, the probability that the two nodes of an edge group are assigned the same number is  $2^{-m}$ , so by a union bound the probability that we successfully pick one node from each group is close to one. Furthermore, we can check in

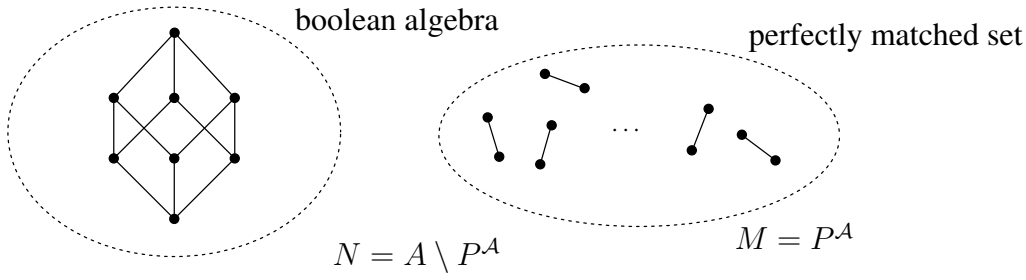


Figure 3.3: The structures in  $\mathcal{B}$  contain a Boolean algebra and a perfectly matched set.

FO whether there is an edge group whose members we can not distinguish, and choose to invariably reject or accept in these cases, resulting in an RFO or co-RFO sentence, respectively.  $\square$

### 3.4.2 BPFO on Ordered Structures is Not Contained in MSO

In the presence of a linear order, *any* query becomes definable in  $L_{\infty\omega}^\omega$ , and the query  $\mathcal{TCTI}$  becomes definable even in FO. However, randomisation adds expressive power to FO also on ordered structures:

**Theorem 37.** *There is a class  $\mathcal{B}$  of ordered structures that is definable in BPFO, but not in MSO.*

Remember that monadic second-order logic MSO is the fragment of second-order logic that allows quantification over individual elements and sets of elements.

Let  $\sigma_{EP\leq} := \{\leq, E, P\}$ , with binary relation symbols  $\leq$  and  $E$ , and a unary predicate  $P$ . We define two classes  $\mathcal{B}'$ ,  $\mathcal{B}$  of  $\sigma_{EP\leq}$ -structures (cf. Figure 3.3):  $\mathcal{B}'$  is the class of all  $\sigma_{EP\leq}$ -structures  $A$  for which

1.  $E(A)$  defines a perfect matching on the set  $M := P(A)$
2. the set  $N := V(A) \setminus P(A)$  forms a Boolean algebra with the relation  $E(A)$  and
3. no  $x \in N$  and  $y \in M$  are  $E$ -related,
4.  $\leq(A)$  is a linear order on the whole structure, which puts the  $M$  before the  $N$  and orders  $M$  in such a way that matched elements are always successive.

It is easy to see that the class  $\mathcal{B}'$  is definable in FO.  $\mathcal{B}$  is the subclass of  $\mathcal{B}'$  whose elements satisfy the additional condition

$$2^{|M|} \geq |N|^2. \quad (3.4)$$

We will prove that  $\mathcal{B}$  is definable in BPFO, but not in MSO. To prove that  $\mathcal{B}$  is definable in BPFO, we will use the following lemma:

**Lemma 38** (Birthday Paradox). *Let  $m, n \geq 1$  and let  $F : [n] \rightarrow [m]$  be a random function drawn uniformly from the set of all such functions.*

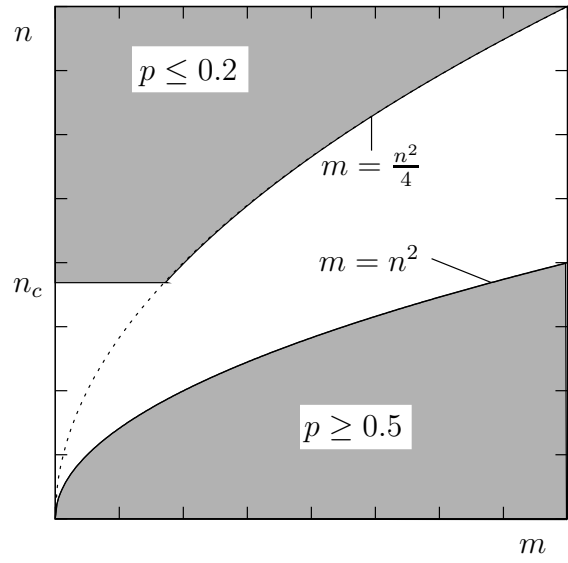


Figure 3.4: The Birthday Paradox with  $\epsilon_1 = 0.2$ ,  $\epsilon_2 = 0.5$  and  $c = 4$ . Here,  $p$  denotes  $\mathbb{P}(F \text{ injective})$ .

1. For any  $\epsilon_1 > 0$  and  $c > 2 \ln \frac{1}{\epsilon_1}$  there is an  $n_c \geq 1$  such that if  $n > n_c$  and  $m \leq \frac{n^2}{c}$  we have

$$\mathbb{P}(F \text{ is injective}) \leq \epsilon_1.$$

2. For any  $\epsilon_2 > 0$ , if  $m \geq \frac{n^2}{2\epsilon_2}$ , then

$$\mathbb{P}(F \text{ is injective}) \geq 1 - \epsilon_2.$$

*Proof.* For the first part, we note that

$$\begin{aligned} \mathbb{P}(F \text{ injective}) &= \prod_{i=0}^{n-1} \left(1 - \frac{i}{m}\right) \\ &\leq \prod_{i=0}^{n-1} \exp\left(-\frac{i}{m}\right) \\ &= \exp\left(-\frac{n(n-1)}{2m}\right). \end{aligned}$$

This is  $\leq \epsilon_1$  if

$$m \leq \frac{n(n-1)}{2 \ln \frac{1}{\epsilon_1}},$$

and for  $c > 2 \ln \frac{1}{\epsilon_1}$  and big enough  $n$  this is the case if

$$m \leq \frac{n^2}{c}.$$

For the second part, note that

$$\begin{aligned} \mathbb{P}(F \text{ not injective}) &= \mathbb{P}\left(\bigcup_{1 \leq i < j \leq n} \{F(i) = F(j)\}\right) \\ &\leq \sum_{i < j} \frac{1}{m} \\ &\leq \frac{n^2}{2m}. \end{aligned}$$

□

*Proof of Theorem 37.* To see that  $\mathcal{B}$  is not definable in **MSO**, we use two simple and well-known facts about **MSO**. The first is that for every  $q \geq 0$  there are natural numbers  $p, m$  such that for all  $k \geq 0$ , a plain linear order of length  $m$  is indistinguishable from the linear order of length  $m + k \cdot p$  by **MSO**-sentences of quantifier rank at most  $q$ . The same fact also holds for linear orders with a perfect matching on successive elements, because such a matching is definable in **MSO** anyway. The second fact we use is a version of the Feferman-Vaught Theorem. Suppose that we have a linearly ordered structure of the form  $A \cup B$ , and the two parts  $A, B$  are disjoint and not related except by the linear order, which puts  $A$  completely before  $B$ . Let  $q \geq 0$  and  $A'$  another linearly ordered structure that is indistinguishable from  $A$  by all **MSO**-sentences of quantifier rank at most  $q$ . Then the structure  $A' \cup B$  is indistinguishable from  $A \cup B$  by all **MSO**-sentences of quantifier rank at most  $q$ . If we put these two facts together, we see that for every  $q \geq 0$  there are  $p, m$  such that for all  $k, n$  the structure  $A \in \mathcal{B}$  with parts  $M, N$  of sizes  $m, n$ , respectively, is indistinguishable from the structure  $A'$  with parts of sizes  $m + k \cdot p$  and  $n$ . We can easily choose  $k, n$  in such a way that  $A \notin \mathcal{B}$  and  $A' \in \mathcal{B}$ .

It remains to prove that  $\mathcal{B}$  is definable in **BPFO**. Consider the sentence

$$\varphi_{\text{inj}} := \forall x \forall y \left( x \dot{=} y \vee Px \vee Py \vee \exists z (Pz \wedge \neg(Rxz \leftrightarrow Ryz)) \right),$$

which states that the random binary relation  $R$ , considered as a function

$$f : \begin{cases} N & \rightarrow 2^M, \\ x & \mapsto \{y \in M \mid Rxy\} \end{cases}$$

from  $N$  to subsets of  $M$  (cf. Figure 3.5), is injective. By the definition of  $R$ , the function  $f$  is drawn uniformly from the set of all such functions. If we fix  $|N|$ , the probability for  $f$  to be injective increases monotonically with  $|M|$ . Furthermore, for every structure in  $\mathcal{B}'$ , the size of  $N$  and  $M$  are a power of two and an even number, respectively. Thus

### 3 Randomised Logics

either

$$2^{|M|} \leq \frac{1}{4} |N|^2 \quad \text{or} \quad 2^{|M|} \geq |N|^2,$$

and this factor of 4 translates into a probability gap for  $\varphi_{\text{inj}}$  in all sufficiently large structures in  $\mathcal{B}'$ , by Lemma 38 with  $\epsilon_1 = 0.2$ ,  $\epsilon_2 = 0.5$  and  $c = 4$ . The remaining finitely many structures in  $\mathcal{B}'$  can be dealt with separately.  $\square$

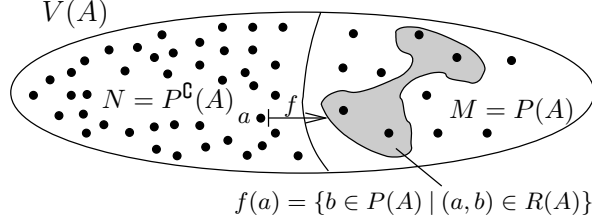


Figure 3.5: The random relation  $R$  interpreted as a function.

#### 3.4.3 RFO is Stronger than FO on Additive Structures

Recall that an additive structure is one whose vocabulary contains a ternary relation  $+$ , such that  $A|_+$  is isomorphic to  $([A], \{(a, b, c) \mid a + b = c\})$ .

**Theorem 39.** *There is a class  $\mathcal{A}$  of additive structures that is definable in RFO and co-RFO, but not in FO.*

Our proof uses the following result:

**Theorem 40** (Lynch [Lyn82]). *For every  $k \in \mathbb{N}$  there is an infinite set  $A_k \subseteq \mathbb{N}$  and a  $d_k \in \mathbb{N}$  such that for all finite  $Q_0, Q_1 \subseteq A_k$  with  $|Q_0| = |Q_1|$  or  $|Q_0|, |Q_1| > d_k$  the structures  $(\mathbb{N}, +, Q_0)$  and  $(\mathbb{N}, +, Q_1)$  satisfy exactly the same FO-sentences of quantifier rank at most  $k$ .*

Here  $(\mathbb{N}, +, Q_i)$  denotes a  $\{+, P\}$ -structure with ternary  $+$  and unary  $P$ , where  $+$  is interpreted as above and  $P$  is interpreted by  $Q_i$ . For a finite set  $M \subseteq \mathbb{N}$  we denote by  $\max M$  the maximum element of  $M$ . By relativising quantifiers to the maximum element satisfying  $P$ , we immediately get the following corollary:

**Corollary 41.** *Let  $k, A_k, d_k, Q_0$  and  $Q_1$  be as above. Then the (finite) structures  $([\max Q_0 + 1], +, Q_0)$  and  $([\max Q_1 + 1], +, Q_1)$  satisfy exactly the same FO-sentences of quantifier rank at most  $k$ .*

We call a set  $Q \subseteq \mathbb{N}$  *sparse* if  $|Q \cap \{n, \dots, 3n\}| \leq 1$  for all  $n \geq 0$ . Note that if  $Q$  is sparse and finite, then  $|Q| \leq \log_3(\max Q) + 1$ . It is easy to see that there is an FO $[\{+, P\}]$ -sentence  $\varphi_{\text{sparse}}$  such that

$$([\max Q + 1], +, Q) \models \varphi_{\text{sparse}} \quad \Leftrightarrow \quad Q \text{ is sparse}$$



for all finite  $Q \subseteq \mathbb{N}$ .

*Proof of Theorem 39.* We define the following class of additive  $\{+, P\}$ -structures:

$$\mathcal{A} = \{([ \max Q + 1 ], +, Q) \mid Q \text{ is finite, sparse and } |Q| \text{ is even}\},$$

with  $+$  defined as usual. It follows immediately from Corollary 41 that  $\mathcal{A}$  is not definable in FO.

It remains to prove that  $\mathcal{A}$  is definable in (co-)RFO. We consider a binary random relation  $R$  on  $\mathcal{Q} = ([ \max Q + 1 ], +, Q)$  for some finite  $Q \subseteq \mathbb{N}$ .

Each element  $a \in [ \max Q + 1 ]$  defines a subset of  $Q$ , namely the set of  $b \in Q$  for which  $(a, b) \in R(\mathcal{Q})$  holds. If  $Q$  is a sparse set, it has

$$2^{|Q|} \leq 2^{\log_3(\max Q)+1} \leq \frac{\max Q}{2 \ln(\max Q)}$$

many subsets, and by standard estimates on the coupon collector's problem (see, e.g., [MR95]; or use a union-bound argument), if  $\max Q$  is large enough, with high probability every subset of  $Q$  is defined by some element of  $[ \max Q + 1 ]$ . We may check in FO whether this is actually the case. If so, we use the random relation  $R$  and the linear order induced by  $+$  to check whether  $Q$  is even. Otherwise we reject (accept) to get an RFO- (co-RFO-)sentence.  $\square$

## Notes

The results in section 3.4 were obtained together with Martin Grohe and appeared in [EG10]. A journal version will appear in [EG11].



## 4 Derandomising Logics

In the previous chapter we have introduced randomised logics and proved that in general they indeed gain expressive power over their non-randomised counterparts. We complement those results in this chapter by proving several derandomisation results for BPFO. By a derandomisation result, we mean a result of the form

$$\text{BPL} \preceq \text{L}',$$

where  $\text{L}'$  is a non-randomised logic. If  $\text{L}' = \text{L}$  we say that BPL can be *fully derandomised*.

As a warm-up, we use the classical 0-1-law for infinitary logic  $\text{L}_{\infty\omega}^\omega$  to prove that, over the empty vocabulary,  $\text{BPL}_{\infty\omega}^\omega$  can be fully derandomised:

**Observation 42.** *Let  $\varphi \in \text{BPL}_{\infty\omega}^\omega[\emptyset]$ . Then there is a  $\psi \in \text{FO}[\emptyset]$  such that*

$$\varphi \equiv \psi.$$

*Proof.* By the 0-1-law for  $\text{L}_{\infty\omega}^\omega$  (see [KV92]), there is an  $n \in \mathbb{N}$  such that for all sets  $M$  of size  $|M| > n$  either

$$M \models \varphi \quad \text{or} \quad M \not\models \varphi$$

holds. The remaining finitely many cases can be defined in FO. □

Similarly, the Sipser-Gács-Lautemann Theorem that  $\text{BPP} \subseteq \Sigma_2^p \cap \Pi_2^p$  implies, by our capturing result for BPP (cf. section 3.3.2) and well known capturing results for the polynomial hierarchy (cf. [EF99]), that  $\text{BPIFP} + \text{C} \preceq \Sigma_2$ . In particular also  $\text{BPFO} \preceq \Sigma_2$ .

In section 4.1, we show that BPFO can be fully derandomised also on structures with a unary vocabulary, as well as on structures with only a single equivalence relation. Using the same techniques, we also give some examples of queries which can not be defined in BPFO. A similar proof technique had previously been used by Shelah [She96] and Boppana and Spencer [BS95] to prove a so-called *very weak 0-1-law* for first-order logic on ordered graphs.

Using entirely different techniques, in section 4.2 we show that, while  $\text{BPFO} \not\preceq \text{MSO}$  on ordered structures (cf. Thm 37),  $\text{BPFO} \preceq \text{MSO}$  on additive structures. Finally, in section 4.3 we prove that a certain query on word models is not definable in BPFO. Whether BPFO can be fully derandomised on word structures (either with an addition relation or just with a successor relation) is an interesting open question, and our non-definability results are a first step towards answering this question.

## 4.1 BPFO $\equiv$ FO on Unary Vocabularies

**Theorem 43.** *Let  $\tau = \{P_1, \dots, P_s\}$  be a vocabulary containing only unary relations, and let  $\varphi \in \text{BPFO}$ . Then there is a (non-randomised)  $\text{FO}[\tau]$ -sentence defining the same query as  $\varphi$ .*

We may restrict ourselves to structures in which every element satisfies exactly one of the  $P_i$ , and we call these  $\tau$ -coloured structures. In fact, a  $\tau$ -structure can be seen as a set partitioned into  $2^s$  classes, where the elements in each class satisfy exactly the same predicates  $P_i$ . We introduce a new vocabulary  $\tau' = \{P'_I \mid I \subseteq [s]\}$  and associate with each  $\tau$ -structure a  $\tau'$ -coloured structure and vice versa in the obvious way. Similarly, each atomic formula  $P_i x$  can be expressed as a boolean combination of atomic formulas  $P'_I x$  and vice versa.

Up to isomorphism, a (finite)  $\tau$ -coloured structure is described uniquely by a tuple  $\vec{n} = (n_1, \dots, n_s) \in \mathbb{N}^s$  of non-negative integers giving the size of each class, and we will denote structures by such tuples. We denote the size of such a structure by  $\|\vec{n}\| := \sum_{i=1}^s n_i$ . For each  $k \in \mathbb{N}$  we define an equivalence relation  $\sim_k$  on  $\mathbb{N}^s$  by saying  $\vec{n} \sim_k \vec{m}$  iff

$$n_i = m_i \quad \text{or} \quad n_i \geq k \text{ and } m_i \geq k$$

for all  $1 \leq i \leq s$ . Then  $\sim_k$  describes exactly the expressive power of first-order sentences of quantifier rank  $k$  on  $\tau$ -coloured structures:

**Lemma 44.** *Let  $\varphi$  be an  $\text{FO}[\tau]$ -sentence of quantifier rank  $\leq k$ . Then on  $\tau$ -coloured structures,  $\text{Mod}(\varphi)$  is a union of  $\sim_k$ -equivalence classes. Conversely, every union of  $\sim_k$ -equivalence classes can be defined by an  $\text{FO}[\tau]$ -sentence of quantifier rank  $\leq k$ .*

*Proof.* This is a standard application of Ehrenfeucht-Fraïssé games, see, e.g., [EF99, ex. 2.3.12].  $\square$

We may thus restate Theorem 43 as follows:

**Lemma 45.** *Let  $\tau = \{P_1, \dots, P_s\}$  be as above and let  $\rho$  be any relational vocabulary with  $\tau \cap \rho = \emptyset$ . Then for every  $\varphi \in \text{FO}[\tau \cup \rho]$  and  $0 < \alpha < \beta < 1$  one of the following holds:*

1. *there is a tuple  $(n_1, \dots, n_s) \in \mathbb{N}^s$  with*

$$\mathbb{P}(A \models \varphi) \in (\alpha, \beta)$$

*or*

2. *there is a  $k \in \mathbb{N}$  such that for all  $\vec{n}, \vec{m}$  with  $\vec{n} \sim_k \vec{m}$  the probabilities  $\mathbb{P}(\vec{n} \models \varphi)$  and  $\mathbb{P}(\vec{m} \models \varphi)$  are either both  $\leq \alpha$  or both  $\geq \beta$ .*

The proof of this lemma is based on the fact that, if we make a large colour class a little smaller by removing one element, the satisfaction probability of an  $\text{FO}[\tau \cup \rho]$ -sentence does not change by much. Here, *large* means both absolutely large (at least a

certain number of elements) and relatively large, i.e., containing at least some inverse polynomial fraction of all elements. This is made precise in the following lemma, which we prove below:

**Lemma 46.** *Let  $\tau = \{P_1, \dots, P_s\}$  and  $\rho$  be vocabularies as above, and  $\varphi \in \text{FO}[\tau \cup \rho]$ . For every  $\epsilon > 0$  and  $c > 1$  there is a  $k = k_{c, \epsilon, \varphi} \in \mathbb{N}$  such that the following holds: If  $\vec{n} \in \mathbb{N}^s$  is a tuple such that  $n_i \geq \|\vec{n}\|^{1/c}$  and  $n_i \geq k$ , then*

$$|\mathbb{P}(\vec{n} \models \varphi) - \mathbb{P}(\vec{n}' \models \varphi)| < \epsilon,$$

where  $n'_i = n_i - 1$  and  $n'_j = n_j$  for  $j \neq i$ .

*Proof of Lemma 45.* Let  $\varphi$  be any  $\text{FO}[\tau \cup \rho]$ -sentence and let  $k = k_{1/s, \beta - \alpha, \varphi}$  be the constant which Lemma 46 yields for  $c = 2$  and  $\epsilon = \beta - \alpha$ . For any tuple  $\vec{n} = (n_1, \dots, n_s) \in \mathbb{N}^s$ , the tuple  $\vec{\nu}$  with

$$\nu_i = \min\{n_i, k\}$$

is a canonical representative of its  $\sim_k$ -equivalence class. We give a sequence

$$\vec{n} = \vec{n}_0, \vec{n}_1, \dots, \vec{n}_l = \vec{\nu}$$

of tuples such that  $\vec{n}_i \sim_k \vec{n}_{i+1}$  and

$$|\mathbb{P}(\vec{n}_i \models \varphi) - \mathbb{P}(\vec{n}_{i+1} \models \varphi)| < \beta - \alpha$$

hold for all  $0 \leq i < l$ . We define such a sequence by successively decreasing one of the maximal entries which are greater than  $k$  until there are no such entries left. Because any maximal entry of a tuple  $\vec{n} \in \mathbb{N}^s$  must be at least  $\|\vec{n}\|/s > \|\vec{n}\|^{1/2}$  for large enough  $\|\vec{n}\|$ , Lemma 46 precisely states that the satisfaction probability of  $\varphi$  never changes by more than  $\beta - \alpha$  in each step, as claimed.

But now the satisfaction probabilities  $\mathbb{P}(\vec{n}_i \models \varphi)$  along the sequence are either all  $\leq \alpha$ , all  $\geq \beta$ , or one of them is in the open interval  $(\alpha, \beta)$ . Because  $\vec{\nu}$  is the same for all tuples in a  $\sim_k$ -equivalence class, the statement of the theorem follows.  $\square$

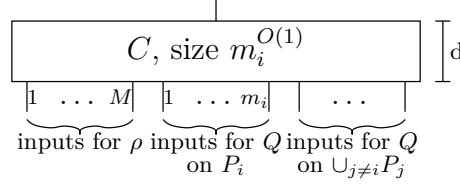
Notice that there may well be  $\vec{n}$  and  $\vec{m}$  with  $\vec{n} \sim_k \vec{m}$  and such that

$$|\mathbb{P}(\vec{n} \models \varphi) - \mathbb{P}(\vec{m} \models \varphi)|$$

is arbitrarily close to 1, but in that case, for every  $\mathbb{P}(\vec{n} \models \varphi) < \alpha < \beta < \mathbb{P}(\vec{m} \models \varphi)$  we can find a  $\vec{u}$  with  $\mathbb{P}(\vec{u} \models \varphi) \in (\alpha, \beta)$ .

*Proof of lemma 46.* We introduce a new unary relation symbol  $Q$  and define an  $\text{FO}[\tau \cup \rho \cup \{Q\}]$ -formula  $\psi$  by restricting all quantifiers of  $\varphi$  to  $Q \cup \bigcup_{j \neq i} P_j$ . That is, we define  $\psi$  recursively from  $\varphi$  by

- if  $\varphi = \exists x \varphi'$  then  $\psi := \exists x (Qx \vee \bigvee_{j \neq i} P_j x) \wedge \psi'$ ,
- if  $\varphi = \forall x \varphi'$  then  $\psi := \forall x ((Qx \vee \bigvee_{j \neq i} P_j x) \rightarrow \psi')$ ,


 Figure 4.1: A polynomial-size, bounded-depth circuit for  $\psi$ 

- if  $\varphi = \neg\varphi'$ , then  $\psi := \neg\psi'$ ,
- if  $\varphi = \varphi' \vee \varphi''$ , then  $\psi := \psi' \vee \psi''$ ,
- if  $\varphi = \varphi' \wedge \varphi''$ , then  $\psi := \psi' \wedge \psi''$ , and
- $\psi := \varphi$  otherwise.

Define  $\vec{m}$  by

$$m_i := 2n_i \quad \text{and} \quad m_j := n_j \text{ for } j \neq i.$$

Treating  $Q$  as a random relation (along with the relations in  $\rho$ ) and conditioning on the size of  $Q \cap P_i$  we get

$$\mathbb{P}(\vec{n} \models \varphi) = \mathbb{P}_{X \in \mathcal{X}(\vec{m}, \rho \cup \{Q\})} (X \models \psi \mid |Q \cap P_i| = n_i)$$

and

$$\mathbb{P}(\vec{n}' \models \varphi) = \mathbb{P}_{X \in \mathcal{X}(\vec{m}, \rho \cup \{Q\})} (X \models \psi \mid |Q \cap P_i| = n_i - 1).$$

Our goal is to show that these two (conditional) probabilities are not too far apart. We first translate the sentence  $\psi$  into a bounded-depth, polynomial-size circuit  $C$  as in Figure 4.1. The depth  $d$  of this circuit is equal to the quantifier depth of  $\psi$ , and it has one input for each relation symbol in  $\rho \cup \{Q\}$  and each tuple of universe elements of appropriate arity. (We assume the unary predicates  $P_1, \dots, P_s$  to be hard-wired into the circuit.) In particular, there are  $m_i = 2n_i$  inputs which determine the set  $Q \cap P_i$ .

The inputs corresponding to  $Q \cap \bigcup_{j \neq i} P_j$  are, by our construction of  $\psi$ , irrelevant and we fix them to 0. Suppose there are  $M$  inputs corresponding to random relations in  $\rho$ . For each way of fixing these inputs to a certain value  $y \in \{0, 1\}^M$  we get a circuit  $C_y$  on  $m_i$  inputs, which is of the same depth as  $C$ . Furthermore, because  $M = \|\vec{n}\|^{O(1)}$  and we assumed  $n_i$  to be  $\geq \|\vec{n}\|^{1/c}$ , the size of  $C_y$  is polynomial in  $m_i$ .

By Theorem 5, the average sensitivity of  $C_y$  is polylogarithmic in  $n_i$ , and therefore also in  $m_i$ . This means that if  $Q \subseteq [m_i]$  and  $q \in [m_i]$  are chosen uniformly and independent of each other, then

$$\mathbb{P}(C_y(Q) \neq C_y(Q \triangle \{q\})) < \frac{(\log m_i)^{O(1)}}{m_i} < m_i^{-0.9}$$

for  $m_i$  large enough. Notice that Boppana's upper bound depends only on the size and depth of the  $C_y$  and thus it is independent of the particular choice of  $y$ .

Let  $A$  be the event that both  $|Q \cap P_i| = n_i$  and  $q \in Q$ . Then

$$\mathbb{P}(A) = \frac{1}{2^{2n_i+1}} \binom{2n_i}{n_i},$$

which is  $\Theta(n_i^{-1/2})$  and therefore  $\Theta(m_i^{-1/2})$  by standard calculations (see, e.g., [Fel57]). By the independence of the inputs of  $C$  we have

$$\mathbb{P}(\vec{n} \models \varphi) = 2^{-M} \sum_y \mathbb{P}(C_y(Q) = 1 \mid A)$$

and

$$\mathbb{P}(\vec{n}' \models \varphi) = 2^{-M} \sum_y \mathbb{P}(C_y(Q \triangle \{q\}) = 1 \mid A)$$

We may now bound the difference of these probabilities as follows:

$$\begin{aligned} |\mathbb{P}(\vec{n} \models \varphi) - \mathbb{P}(\vec{n}' \models \varphi)| &\leq 2^{-M} \sum \mathbb{P}(C_y(Q) \neq C_y(Q \triangle \{q\}) \mid A) \\ &\leq 2^{-M} \sum \frac{\mathbb{P}(C_y(Q) \neq C_y(Q \triangle \{q\}) \cap A)}{\mathbb{P}(A)} \\ &\leq 2^{-M} \sum \frac{\mathbb{P}(C_y(Q) \neq C_y(Q \triangle \{q\}))}{\mathbb{P}(A)} \\ &\leq m_i^{-0.9} \cdot \Theta(m_i^{1/2}) < m_i^{-0.3} \end{aligned}$$

for  $m_i$  large enough. We assumed  $m_i \geq k$ , and thus this difference is  $< \epsilon$  if we choose  $k$  large enough.  $\square$

The above proof technique can be adapted to yield the following somewhat stronger result:

**Theorem 47.** *Let  $\sigma = \{E\}$  be a vocabulary containing just one binary relation  $E$ , and let  $\mathcal{EQ}$  be the class of all finite structures  $A$  for which  $E(A)$  is an equivalence relation. Then BPFO = FO on  $\mathcal{EQ}$ .*

**Remark 48.** Note that because  $\mathcal{EQ}$  is definable in FO, for every sentence  $\varphi$  with a probability gap on  $\mathcal{EQ}$  there is a sentence  $\varphi'$  which is equivalent to  $\varphi$  on  $\mathcal{EQ}$  and has a probability gap on all finite structures.

*Proof.* Up to isomorphism, a structure  $A \in \mathcal{EQ}$  is determined by a function  $f^A : \mathbb{N} \rightarrow \mathbb{N}$  such that  $f^A(s)$  counts the number of equivalence classes of size  $s$  (so that  $|V(A)| = \sum s f^A(s) =: \|f\|$ ). For each  $k \in \mathbb{N}$  we define a function

$$f_k^A(s) = \begin{cases} \min\{k, f^A(s)\} & \text{if } s < k, \\ \min\{k, \sum_{i \geq k} f^A(i)\} & \text{if } s = k, \\ 0 & \text{if } s > k. \end{cases}$$

We say  $A \sim_k B$  if  $f_k^A(s) = f_k^B(s)$  for all  $s \in \mathbb{N}$ . By standard techniques, a query  $\mathcal{Q} \subseteq \mathcal{EQ}$  is definable in FO iff it is a union of  $\sim_k$ -equivalence classes for some  $k$ . A function  $f$  is  $k$ -canonical if  $f(s) \leq k$  for all  $s$  and  $f(s) = 0$  for all  $s > k$ . The  $k$ -canonical functions form a system of representatives for the equivalence relation  $\sim_k$ , and we denote the representative equivalent to  $f$  by  $\tilde{f}$ .

For notational convenience, again we assume there is only one random relation symbol  $R$ . Fix a formula  $\varphi \in \{E, R\}$  and an  $\epsilon > 0$ . As in Lemma 46 we show that there is a  $k$  such that for every  $f$  there is a sequence

$$f = f_0 \sim_k f_1 \sim_k f_2 \sim_k \cdots \sim_k f_l = \tilde{f}$$

with  $|\mathbb{P}(f_i \models \varphi) - \mathbb{P}(f_{i+1} \models \varphi)| < \epsilon$  along the sequence. To get from  $f_i$  to  $f_{i+1}$  we proceed as follows: Suppose  $n := \|f_i\| > k^3$ . If one equivalence class has  $> n^{1/3}$  elements (i.e.  $f_i(s) > 0$  for some  $s > n^{1/3}$ ) we remove one element from that class. Otherwise, there must be an  $s \leq n^{1/3}$  such that  $f_i(s) > n^{1/3}$ . In this case, remove an entire equivalence class of size  $s$ . Finally, if  $\|f_i\| \leq k^3$ , we may remove elements from equivalence classes of size  $> k$  and remove an equivalence class of size  $s$  if there are more than  $k$  classes of that size. Proceeding in this way we eventually reach  $\tilde{f}$ .

Removing an element from a class is done by randomly choosing from a class of twice the size, and removing a class of a certain size is done by randomly choosing among twice as many classes of that size.  $\square$

Using the same techniques as in the proof of Theorem 43, we obtain the following non-definability results:

**Theorem 49.** *The following queries on finite structures are not definable in BPFO:*

- (a) *Over the vocabulary  $\{\leq\}$  containing a binary relation symbol  $\leq$ , the query “ $\leq$  defines a linear order of even cardinality”*
- (b) *Over the vocabulary  $\{E\}$  containing a binary relation symbol  $E$ , the query “ $E$  defines a connected graph”*
- (c) *Over the vocabulary  $\{+1\}$  containing a binary relation symbol  $+1$ , the query “initial segment of the natural numbers, treating  $+1$  as a successor relation”.*

*Proof.* Denote by  $\mathcal{O}_n$  the linear order on  $n$  elements. For query (a), introduce a new random unary relation  $P$  on a total linear order of length  $2n$  and relativise all quantifiers to  $P$  as in the proof of Theorem 43. Letting  $n$  tend to infinity, this shows that

$$\left| \mathbb{P}_{X \in \mathcal{X}(\mathcal{O}_n, \rho)}(X \models \varphi) - \mathbb{P}_{X \in \mathcal{X}(\mathcal{O}_{n-1}, \rho)}(X \models \varphi) \right| \rightarrow 0$$

for any FO $[\{\leq\} \cup \rho]$ -sentence  $\varphi$ . Note that this is basically the statement of Shelah’s very weak 0-1-law [She96], but we allow the random part of the structure to be an arbitrary relation, not just the edge-relation of a simple graph. However, Boppana and Spencer’s proof for Shelah’s result [BS95] extends to this case essentially unchanged.



Non-definability of queries (b) and (c) follows because we can define a graph on  $\mathcal{O}_n$  in FO which is connected iff  $n$  is odd. To be precise, define a formula  $\varphi_{+1}$  by

$$\begin{aligned} \varphi_{+1}(x, y) := & \exists z (x \leq z \wedge z \leq y \wedge \neg x \dot{=} z \wedge \neg z \dot{=} y \wedge \forall w (w \leq x \vee z \dot{=} w \vee y \leq w)) \vee \\ & (\forall z z \leq x \wedge \exists z (z \leq y \wedge \neg z \dot{=} y \wedge \forall w (w \dot{=} z \vee y \leq w))) \end{aligned}$$

and use it to define a successor relation  $+1$  in an ordered structure. Identifying the elements of the linear order with the first  $n$  natural numbers, this way we define the successor of element  $x$  to be  $x + 2$ , and additionally the successor of the last element is the second element. This defines an initial segment of the natural numbers iff the size of the linear order is odd.

Similarly, by setting

$$\varphi_E(x, y) := \varphi_{+1}(x, y) \vee \varphi_{+1}(y, x)$$

we define a graph on the linear order which connects elements

- $x$  and  $x + 2$  for all  $1 \leq x \leq n - 2$ ,
- $2$  and  $n - 1$ ,

and this graph is connected iff  $n$  is odd. Thus a BPFO-sentence defining connected graphs could be used to define evenness of a linear order. This argument is essentially taken from [EF99].  $\square$

## 4.2 BPFO is Contained in MSO on Additive Structures

The result of this section complements the result of section 3.4.2 by saying that, on additive structures, every BPFO-sentence is equivalent to an MSO-sentence. That is, we prove:

**Theorem 50.** *Let  $\tau$  be a finite relational vocabulary containing a ternary relation  $+$  and let  $\varphi$  be a BPFO $[\tau]$ -sentence. Then there exists an MSO-sentence  $\psi$  such that on additive structures  $A$*

$$A \models \varphi \quad \Leftrightarrow \quad A \models \psi.$$

We first use Nisan's pseudorandom generator for constant depth circuits [Nis91] to reduce the number of random bits to  $\log^{O(1)} n$ ; throughout this section,  $n$  will denote the size of the input structure. We then derandomise the resulting formula following Lautemann's argument in [Lau83].

In MSO $[+]$ , one can define a multiplication relation (see [Sch06, Lemma 5.4]) and thus quantify over pairs of elements in  $[0, \sqrt{n}]$ . We only need the existence of such a pairing function, a slightly weaker form of which is made precise in the following lemma:

**Lemma 51** (Pairing Lemma). *There are MSO $[+]$ -formulas  $\varphi_p(x)$  and  $\varphi_{\langle \cdot, \cdot \rangle}(x, y, z, w)$  such that on additive structures  $A$*

#### 4 Derandomising Logics

- $\varphi_p(x)$  defines a number  $p$  satisfying

$$\frac{\sqrt{|A|}}{2} \leq p \leq \sqrt{|A|}.$$

Moreover,  $p$  is a prime number.

- For every  $b, c < p$  there is a unique  $m$  such that  $\varphi_{\langle \cdot, \cdot, \cdot \rangle}(0, b, c, m)$  is satisfied. Furthermore, for every  $m$  there is a unique tuple  $(a, b, c) \in [0, p-1]^3$  such that  $\varphi_{\langle \cdot, \cdot, \cdot \rangle}(a, b, c, m)$  is satisfied. Henceforth we write  $m = \langle a, b, c \rangle$  for this.

*Proof.* In  $\text{MSO}[+]$ , we may define a formulas  $\varphi_{X=\langle x \rangle}(X, x)$  and  $\varphi_{\text{divides}}(x, y)$  stating that  $X$  is the set of multiples of  $x$  and  $x$  divides  $y$ , respectively. We may thus check whether  $x$  is a prime number. Furthermore, we may define the set of powers of a prime number  $x$ : It is the largest set containing only numbers whose only prime divisor is  $x$ .

Then  $p$  is the largest prime number whose set of powers contains at least one element other than 0 and itself. Any number  $m \in [0, p^2 - 1]$  may be written as  $m = bp + c$  with  $b, c \in [0, p-1]$ . Both  $b$  and  $c$  are definable in  $\text{MSO}[+]$ ; notice that  $b$  is the largest divisor of  $m - c$  smaller than  $p$ , or 0 if  $m < p$ . For  $m \geq p^2$  we define  $m = \langle a, b, c \rangle$  with  $a \in \{1, 2, 3\}$  and  $m - ap^2 = \langle 0, b, c \rangle$ .  $\square$

Whenever we write  $p$  in this section, we mean the  $p$  defined by the  $\varphi_p$  above. The Pairing Lemma allows us to quantify over binary relations on  $[0, p-1] \cong \mathbb{F}_p$ . In particular, we may define addition and multiplication modulo  $p$ , i.e., there are  $\text{MSO}[+]$ -formulas  $\varphi_+(x, y, z)$  and  $\varphi_\times(x, y, z)$  such that for  $a, b, c \in \mathbb{F}_p$ ,

$$A \models \varphi_+(a, b, c) \iff a + b \equiv c \pmod{p}$$

and

$$A \models \varphi_\times(a, b, c) \iff a \cdot b \equiv c \pmod{p}.$$

For the proof of Theorem 50 we may assume that the BPFO-sentence  $\varphi$  contains only one random relation, say  $R$  of arity  $r$ . We first apply a result by Nisan [Nis91] to reduce the number of random bits:

**Lemma 52.** *For every  $r, d \in \mathbb{N}$  and  $\epsilon > 0$  there are  $\text{MSO}[+]$ -formulas  $\varphi_l(x)$  and  $\varphi_{\text{prg}}(S, x_1, \dots, x_r)$ , where  $S$  is a set variable, such that*

- $\varphi_l$  defines a number  $l \leq \log^{O(1)} n$  and
- if  $\varphi$  is an  $\text{FO}[\tau \cup \{R\}]$ -sentence of quantifier rank  $\leq d$ , where  $\tau$  is some finite relational vocabulary and  $R$  is of arity  $r$ , then

$$\left| \mathbb{P}_{X \in \mathcal{X}(A, \{R\})} (X \models \varphi) - \mathbb{P}_{S \subseteq [l]} (A \models \varphi'(S)) \right| < \epsilon,$$

where  $\varphi'$  is the  $\text{MSO}[+]$ -formula obtained from  $\varphi$  by replacing every occurrence of  $R\vec{x}$  by  $\varphi_{\text{prg}}(S, \vec{x})$ .

## 4.2 BPFO is Contained in MSO on Additive Structures

*Proof.* For any fixed structure  $A$  of size  $n$  we may construct a polynomial-sized circuit  $C_{\varphi,A}$  of depth  $\leq d$  which describes the behaviour of  $\varphi$  on  $(\tau \cup \{R\})$ -expansions of  $A$ . The circuit has  $n^r$  inputs indexed by the elements of  $V(A)^r$ , and an input vector  $\vec{x}$  denotes the  $(\tau \cup \{R\})$ -expansion  $B_{\vec{x}}$  of  $A$  given by

$$\vec{a} \in R(B_{\vec{x}}) \quad \text{iff} \quad x_{\vec{a}} = 1.$$

Then  $C_{\varphi,A}(\vec{x})$  evaluates to 1 iff  $B_{\vec{x}} \models \varphi$ .

Nisan [Nis91] gave a pseudorandom generator for such circuits which hinges on the following lemma:

**Lemma 53** (restated from [Nis91, Lemma 2.2]). *Let  $\{C_n\}$  be a family of circuits of depth  $d$  and polynomial size, let  $m = m(n) = (\log n)^{d+3}$ ,  $l = l(n)$  and suppose for each  $n$  the sets  $A_1^{(n)}, \dots, A_n^{(n)} \subseteq [l]$  satisfy*

- $|A_i^{(n)}| = m$  for all  $1 \leq i \leq n$  and
- $|A_i^{(n)} \cap A_j^{(n)}| \leq \log n$  for all  $1 \leq i \neq j \leq n$ .

Then

$$|\mathbb{P}(C_n(\vec{x}) = 0) - \mathbb{P}(C_n(\oplus_{i \in A_1} y_i, \dots, \oplus_{i \in A_n} y_i) = 0)| \leq \frac{1}{n^c}$$

for any  $c \in \mathbb{N}$  and large enough  $n$ . Here, the first probability is taken uniformly over all strings  $\vec{x} \in \{0, 1\}^n$ , whereas the second is taken uniformly over all strings  $\vec{y} \in \{0, 1\}^l$ .

The resulting pseudorandom generator is depicted in Figure 4.2. Families of sets  $A_i^{(n)}$  satisfying the above conditions are called *partial- $(\log n, m)$ -designs*. Nisan gives a construction with  $l = m^2 = \log^{O(1)} n$ , which drastically reduces the size of the probability space, i.e., the number of random bits needed. We now show how his construction can be defined in  $\text{MSO}[+]$ .

On  $[0, p-1]$ , we may define a formula  $\varphi_{\log}(x, y)$  which is satisfied iff  $x = \lceil \log_2 y \rceil$ . Using this and the fact that

$$2\lceil \log p \rceil - 1 \leq \lceil \log n \rceil \leq 2\lceil \log p \rceil + 2,$$

we let  $\varphi_m(x)$  and  $\varphi_l(x)$  be two formulas defining natural numbers  $m$  and  $l$  such that

- $m$  is a prime number between  $(r^2 \lceil \log n \rceil)^{d+3}$  and  $2(r^2(\lceil \log n \rceil + 3))^{d+3}$
- $l = m^2$ .

Using the pairing function  $\varphi_{\langle \cdot, \cdot \rangle}$  we may assume that  $R$  is a  $3r$ -ary relation which we only need to define for elements in  $\mathbb{F}_p$ . That is, we define  $\varphi_{\text{prg}}(S, x_1, \dots, x_r)$  by

$$\exists z_1 \cdots \exists z_{3r} x_1 = \langle z_1, z_2, z_3 \rangle \wedge \dots \wedge x_r = \langle z_{3r-2}, z_{3r-1}, z_{3r} \rangle \wedge \varphi'_{\text{prg}}(S, z_1, \dots, z_{3r}).$$

The formula  $\varphi'_{\text{prg}}(S, \vec{z})$  takes the parity of a subset of  $S$  indexed by  $\vec{z}$ :

$$\varphi'_{\text{prg}}(S, \vec{z}) := “|S \cap \psi(A; \vec{z})| \text{ is even}”,$$

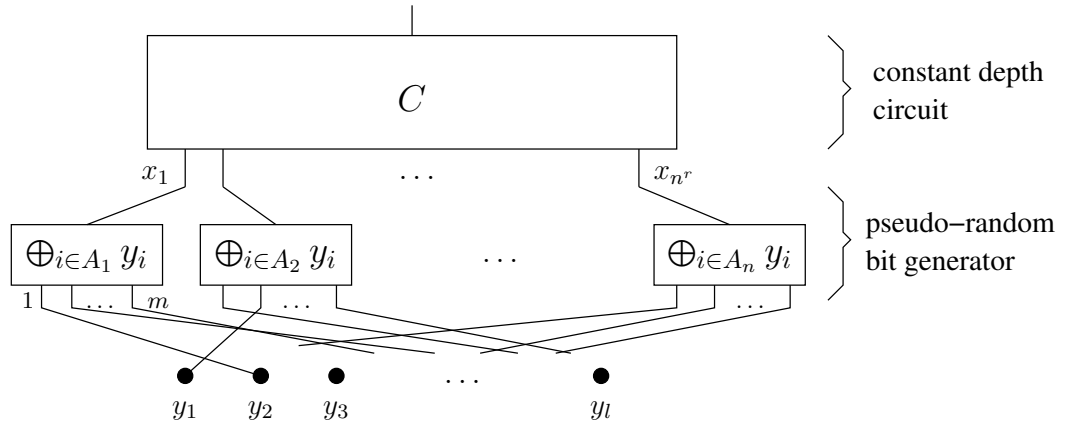


Figure 4.2: Nisan's pseudo-random bit generator. The sets  $A_i \subseteq \{1, \dots, l\}$  form a partial- $(\log n, m)$ -design, i.e., they satisfy  $|A_i| = m$  and  $|A_i \cap A_j| \leq \log n$  for all  $1 \leq i \neq j \leq n$ .

where  $\psi(x, \vec{z})$  is an  $\text{MSO}[+]$ -formula and  $\psi(A; \vec{z}) := \{x \mid A \models \psi(x, \vec{z})\}$ ; evenness may be expressed in  $\text{MSO}$  on ordered structures. By Lemma 53, we are done if we can define a formula  $\psi(x, \vec{z})$  such that

- (i)  $\psi(A; \vec{z}) \subseteq [l]$  for all  $\vec{z} \in \mathbb{F}_p^{3r}$ ,
- (ii)  $|\psi(A; \vec{z})| = m$  for all  $\vec{z} \in \mathbb{F}_p^{3r}$ , and
- (iii)  $|\psi(A; \vec{z}_1) \cap \psi(A; \vec{z}_2)| \leq \log n$  for all  $\vec{z}_1 \neq \vec{z}_2 \in \mathbb{F}_p^{3r}$ ,

which means the sets  $\psi(A; \vec{z})$  form a partial- $(\log n, m)$ -design. We use the same construction as Nisan: We interpret the tuple  $\vec{z}$  as a polynomial  $f_{\vec{z}} \in \mathbb{F}_m[\xi]$  of degree  $\leq \log n$ . The set  $\psi(A; \vec{z})$  is then the graph of this polynomial, namely

$$\psi(A; \vec{z}) = \{(\xi, f_{\vec{z}}(\xi)) \mid \xi \in \mathbb{F}_m\} \subseteq \mathbb{F}_m^2,$$

and we identify  $\mathbb{F}_m^2$  with  $[l]$ . We first encode the coefficients of  $f_{\vec{z}}$  into a set variable  $X$  as follows: Consider the binary representations

$$z_i = \sum_{j \geq 0} z_{i,j} 2^j \quad \text{with } z_{i,j} \in \{0, 1\}$$

of the  $z_i$ . We can define an  $\text{MSO}[+]$ -sentence  $\varphi_{\text{pack}}(\vec{z}, X)$  which holds iff  $X$ , interpreted as a binary relation over  $\mathbb{F}_p$ , holds exactly for pairs  $(a, b)$  with

$$0 \leq a \leq \lceil \log p \rceil \quad \text{and} \quad b = \sum_{1 \leq i \leq 3r} z_{i,a} 2^{i-1}.$$

Thus for each  $0 \leq a \leq \lceil \log p \rceil$  there is exactly one  $b = b(a)$  with  $(a, b) \in X$ , and all  $b$ s are between 0 and  $2^{3r}$ , and thus in  $\mathbb{F}_m$  if  $n$  is large enough. We may now define an

MSO[+]-sentence  $\varphi_{\text{eval}}(X, u, v)$  which, for these  $X$ s, holds iff

$$v = f_{\vec{y}}(u) = \sum_{0 \leq a < \lceil \log p \rceil} b(a)u^a,$$

with addition and multiplication according to  $\mathbb{F}_m$ . Putting these ingredients together, we define

$$\psi(x, \vec{z}) = \exists X \exists u \exists v "0 \leq u, v < m" \wedge \varphi_{\text{pack}}(\vec{z}, X) \wedge \varphi_{\text{eval}}(X, u, v) \wedge "x = u \cdot m + v",$$

which is easily verified to satisfy conditions (i) to (iii) above.  $\square$

So far we have reduced the number of random bits from  $n^r$  to  $l = \log^{O(1)} n$ , and these are conveniently packed into the first  $l$  bits of a single set variable  $S$ . We may now follow Lautemann's proof [Lau83] to derandomise this sentence.

*Proof of Theorem 50.* After applying Lemma 52 we are left with MSO[+]-sentences  $\varphi_l$  and  $\varphi'$  such that  $\varphi_l$  defines a number  $l \leq \log^{O(1)} n$  and  $\varphi'$  has a free set variable  $S$ . We may assume that for all additive structures  $A$ ,

$$\text{either } \mathbb{P}_{S \subseteq [l]} (A \models \varphi'(S)) < \frac{1}{l} \quad \text{or } \mathbb{P}_{S \subseteq [l]} (A \models \varphi'(S)) > 1 - \frac{1}{l}, \quad (4.1)$$

because otherwise we may use independent repetition and majority vote to obtain these bounds. To be precise, let  $\chi(S, i, j)$  be defined by

$$\chi(S, i, j) := (0 \leq i < l) \wedge (0 \leq j < l) \wedge \exists z (z \doteq i \cdot l + j \wedge Sz).$$

That is, we divide the first  $l^2$  bits of  $S$  into  $l$  blocks of  $l$  bits each, and let  $\chi(S, i, j)$  select the  $i$ -th bit of the  $j$ -th block. We replace each occurrence of  $Sx$  in  $\varphi'$  by  $\chi(S, i, x)$  to obtain a formula  $\tilde{\varphi}'(S, i)$ . Because  $l$  is of order  $\log^{O(1)} n$ , we may quantify over pairs of elements of  $[0, l-1]$ , which allows us to express the formula

$$\bar{\varphi}'(S) = "\tilde{\varphi}'(S, i) \text{ holds for at least half of the } i \in [0, l-1]"$$

in MSO[+], e.g., by stating that there exists a matching  $M$  on  $[0, l-1]$  such that

- if  $\{i, j\} \in M$ , then exactly one of  $\tilde{\varphi}'(S, i)$  and  $\tilde{\varphi}'(S, j)$  holds and
- all  $i \in [0, l-1]$  for which  $\tilde{\varphi}'(S, i)$  does not hold are matched by  $M$ .

Then  $\bar{\varphi}'$  uses  $l^2 = \log^{O(1)} n$  many bits of  $S$ , and by the Chernoff bound on the tails of the binomial distribution it satisfies (4.1), even with  $l$  replaced by  $l^2$  (details can be found in [AB09, sec. 7.4]).

We identify subsets of  $[l]$  with vectors in  $\mathbb{F}_2^l$ . Let  $M \subseteq \mathbb{F}_2^l$  be the set of vectors for

which  $A \models \varphi'(S)$  holds. Equation (4.1) translates into

$$|M| < \frac{|\mathbb{F}_2^l|}{l} \quad \text{or} \quad |M| > \left(1 - \frac{1}{l}\right) |\mathbb{F}_2^l|.$$

For a vector  $\vec{y} \in \mathbb{F}_2^l$  we define

$$\vec{y} \oplus M := \{\vec{x} \oplus \vec{y} \mid \vec{x} \in M\}$$

to be the set  $M$  translated by  $\vec{y}$ . We claim the following:

- (a) If  $|M| < |\mathbb{F}_2^l|/l$ , then for every choice of vectors  $\vec{y}_1, \dots, \vec{y}_l$  we have

$$\bigcup_{1 \leq i \leq l} (\vec{y}_i \oplus M) \neq \mathbb{F}_2^l.$$

- (b) If  $|M| > (1 - 1/l) |\mathbb{F}_2^l|$ , then there are vectors  $\vec{y}_1, \dots, \vec{y}_l$  such that

$$\bigcup_{1 \leq i \leq l} (\vec{y}_i \oplus M) = \mathbb{F}_2^l.$$

The first claim follows immediately from  $|\vec{y} \oplus M| = |M|$ . For (b), assume that we randomly choose the vectors  $\vec{y}_i$  independently and uniformly from  $\mathbb{F}_2^l$ . For any vector  $\vec{x} \in \mathbb{F}_2^l$  we have

$$\begin{aligned} \mathbb{P}\left(\vec{x} \notin \bigcup (\vec{y}_i \oplus M)\right) &= \prod_i \mathbb{P}(\vec{x} \notin \vec{y}_i \oplus M) \\ &\leq \left(\frac{1}{l}\right)^l, \end{aligned}$$

by the independence of the  $\vec{y}_i$ . But then the expected number of vectors *not* in  $\bigcup (\vec{y}_i \oplus M)$  is

$$\begin{aligned} \mathbb{E}\left[\left|\mathbb{F}_2^l \setminus \bigcup (\vec{y}_i \oplus M)\right|\right] &= \sum_{\vec{x} \in \mathbb{F}_2^l} \mathbb{P}\left(\vec{x} \notin \bigcup (\vec{y}_i \oplus M)\right) \\ &\leq \frac{|\mathbb{F}_2^l|}{l^l} = \left(\frac{2}{l}\right)^l < 1, \end{aligned}$$

so there must be a choice of  $\vec{y}_i$ s such that this number is zero, i.e.,  $\bigcup (\vec{y}_i \oplus M) = \mathbb{F}_2^l$ .

Again using the formula  $\chi(S, i, j)$ , we can pack the vectors  $\vec{y}_1, \dots, \vec{y}_l$  into a single existentially quantified set variable and check that  $\bigcup (\vec{y}_i \oplus M) = \mathbb{F}_2^l$  as follows:

$$\varphi'' = \exists Y \forall X \exists i \varphi'(X \oplus \chi(Y, i, \cdot)),$$

where  $\varphi'(X \oplus \chi(Y, i, \cdot))$  is the formula  $\varphi'(S)$  with every occurrence of  $Sx$  replaced by

$$(Xx \wedge \chi(Y, i, x)) \vee (\neg Xx \wedge \neg \chi(Y, i, x)).$$

Claims (a) and (b) imply that

$$A \models \varphi'' \Leftrightarrow \mathbb{P}(A \models \varphi'(S)) > 1 - \frac{1}{l},$$

which completes the proof.  $\square$

### 4.3 Randomised First-Order Logic on Words

We denote by  $\text{FO}[+1]$ ,  $\text{FO}[\leq]$ ,  $\text{BPFO}[+1]$ , and  $\text{BPFO}[\leq]$  (randomised) first-order logic restricted to word models of the appropriate type. There are two natural definitions of BPFO on restricted classes of structures, namely one which demands BPFO sentences to have a gap on *all* finite structures, and one which demands this only on structures from the restricted class. Because the fact that  $\leq$  defines a linear order is definable in FO, word models of the second type can be defined in FO and this distinction does not affect the expressive power of  $\text{BPFO}[\leq]$ . In contrast to this, the successor relation  $+1$  can not be defined in FO, because connexness of the transitive closure of  $+1$  is not definable. By Theorem 49(c), this holds true also for BPFO. Therefore, the two definitions of  $\text{BPFO}[+1]$  potentially have different expressive power. Our counterexample in Theorem 54 works for both variants.

The expressive power of  $\text{FO}[+1]$  and  $\text{FO}[\leq]$  is well understood, see [Str94]. In particular, the query

$$Q := a^*ba^*ca^* \subseteq \{a, b, c\}^*$$

of all words which contain exactly one  $b$  to the left of exactly one  $c$  and an arbitrary number of  $a$ s is not definable in  $\text{FO}[+1]$ . It is easily seen to be definable in  $\text{FO}[\leq]$  by the sentence

$$\exists x \exists y (P_b x \wedge P_c y \wedge x \leq y \wedge \forall z (P_a z \vee z \dot{=} x \vee z \dot{=} y)).$$

We show that  $Q$  is not definable in  $\text{BPFO}[+1]$ :

**Theorem 54.** *There is no  $\text{BPFO}[+1]$ -sentence  $\varphi$  such that*

$$w \models \varphi \Leftrightarrow w \in Q$$

*for all  $w \in \{a, b, c\}^*$ .*

For the proof we will use Ehrenfeucht-Fraïssé games to show that certain structures can not be distinguished by first-order formulas of a given quantifier rank. Two structures  $A$  and  $B$  are called  $m$ -equivalent, written  $A \equiv_m B$ , if they satisfy exactly the same FO-formulas of quantifier rank up to  $m$ . By Ehrenfeucht's Theorem (cf. [EF99]), this is equivalent to the existence of a winning strategy for Duplicator in the following game (called Ehrenfeucht-Fraïssé game):

#### 4 Derandomising Logics

Two players, called Spoiler and Duplicator, take turns in choosing elements from two structures  $A$  and  $B$ . Spoiler moves first. If, in the  $k$ -th round, Spoiler chooses an element  $a_k$  from structure  $A$ , Duplicator has to answer with an element  $b_k$  from structure  $B$ , and vice versa. Duplicator wins if, after  $m$  rounds have been played,  $a_1 \dots a_m \mapsto b_1 \dots b_m$  is a partial isomorphism. Therefore, by exhibiting a winning strategy for Duplicator in the  $m$ -round Ehrenfeucht-Fraïssé game, one can show that  $A$  and  $B$  are  $m$ -equivalent.

*Proof of theorem 54.* Let  $\sigma = \{+1, P_a, P_b, P_c\}$  be the vocabulary of our word models. We show the theorem by exhibiting a sequence of pairs of words  $v_n, w_n$  such that

- (i)  $v_n \in Q, w_n \notin Q$  for all  $n \geq 1$  and
- (ii) for every vocabulary  $\rho$  disjoint from  $\sigma$  and every  $\text{FO}[\sigma \cup \rho]$ -sentence  $\varphi$ ,

$$|\mathbb{P}(v_n \models \varphi) - \mathbb{P}(w_n \models \varphi)| \rightarrow 0 \quad (n \rightarrow \infty).$$

In fact, choosing

$$\begin{aligned} v_n &= a^n b a^n c a^n \\ w_n &= a^n c a^n b a^n \end{aligned}$$

will do. Condition (i) is obviously satisfied. For condition (ii), let  $\rho$  be disjoint from  $\sigma$  and let  $\varphi$  be a sentence of quantifier rank  $r$ . The successor relation induces a distance measure on the elements of the structures, which we denote by  $d$ ; we assume  $d(x, y) = 1$  if  $x = y + 1$  or  $y = x + 1$ . We denote by  $d_r$  the bounded distance function

$$d_r(x, y) := \begin{cases} d(x, y) & \text{if } d(x, y) \leq r \\ \infty & \text{otherwise.} \end{cases}$$

By  $S^r(x)$  we denote the  $r$ -ball around an element  $x$  in (a  $(\sigma \cup \rho)$ -expansion of) a word structure  $A$ , i.e.,

$$S^r(x) := \{y \in V(A) \mid d(x, y) \leq r\},$$

and if  $a_1, \dots, a_k$  are elements of  $V(A)$ , then  $A|_{S^r(a_1, \dots, a_k)}$  denotes the induced substructure of  $A$  on the union  $\bigcup_{i=1}^k S^r(a_i)$  of the  $r$  balls around these elements. We say that two sets  $U, V \subseteq V(A)$  *touch* if there are  $x \in U$  and  $y \in V$  with  $x = y + 1$  or  $y = x + 1$ .

For  $n > 3^r$ , the word structures  $v_n$  and  $w_n$  satisfy exactly the same first-order sentences of quantifier rank up to  $r$ . A winning strategy for the  $r$ -move Ehrenfeucht-Fraïssé game on  $v_n$  and  $w_n$  can be given explicitly as follows: For ease of notation, we denote the first and the last position of  $v_n$  by  $a_1$  and  $a_2$ , the unique position containing a  $b$  by  $a_3$  and that containing a  $c$  by  $a_4$ , and likewise for  $b_1, \dots, b_4$ . Suppose after  $k$  moves, elements  $a_5, \dots, a_{k+4}$  have been chosen in  $v_n$ , and elements  $b_5, \dots, b_{k+4}$  have been chosen in  $w_n$ . Assume Spoiler chooses an element  $a$  in  $v_n$ . Throughout the game, Duplicator maintains the property that

$$d_{3^r-k}(a_i, a_j) = d_{3^r-k}(b_i, b_j) \tag{4.2}$$



for  $1 \leq i, j \leq k+4$ . Notice that this property holds before the first move (i.e., for  $a_1, \dots, a_4$  and  $b_1, \dots, b_4$ ) if  $n > 3^r$ . Let  $r' = r - k - 1$  be the number of rounds remaining after the  $k$ -th move.

- (I) If  $a$  is in  $v_n|_{S^{3^{r'}}(a_1, \dots, a_{k+4})}$ , then choose the corresponding element in  $w_n$ , i.e., the unique element  $b \in V(w_n)$  which has

$$d_{3^{r'}}(a_i, a) = d_{3^{r'}}(b_i, b)$$

for  $1 \leq i \leq k+4$ . This is possible because if  $d(b_i, b), d(b_j, b) \leq 3^{r'}$ , then  $d(b_i, b_j) \leq 2 \cdot 3^{r'} < 3^{r-k}$  and  $d_{3^{r-k}}(a_i, a_j) = d_{3^{r-k}}(b_i, b_j)$  by property (4.2).

- (II) Otherwise, choose any element of  $w_n$  which has distance  $> 3^{r'}$  from all elements  $b_1, \dots, b_{k+4}$ .

Duplicator's answer if Spoiler chooses an element  $b$  in  $w_n$  is determined analogously. After  $r$  rounds have been played, the map  $a_i \mapsto b_i$  is a partial isomorphism, because all relations in  $\sigma$  are determined by  $d_1$ -distances. This is because on the words  $v_n$  and  $w_n$ , the relations  $P_a$ ,  $P_b$  and  $P_c$  depend only on the  $d_1$ -distance from  $u$  and  $v$ , which are parts of the tuples.

We now extend this strategy to random expansions  $X$  of  $v_n$  and  $Y$  of  $w_n$ . Let

$$\begin{aligned} c_0 &:= 1, \\ c_{i+1} &:= 4r_i + 2. \end{aligned}$$

In the game on  $X$  and  $Y$ , Duplicator maintains the stronger property that after the  $k$ -th move,

$$X_k := X|_{S^{c_{r-k}}(a_1, \dots, a_{k+4})} \cong Y|_{S^{c_{r-k}}(b_1, \dots, b_{k+4})} =: Y_k, \quad (4.3)$$

treating the  $a_i$ s and  $b_i$ s as constants. That this, there is an isomorphism  $f : X_k \xrightarrow{\sim} Y_k$  such that  $f(a_i) = b_i$  for  $1 \leq i \leq k+4$ . This is of course not possible for all random expansions: At the very least, the random expansions have to agree on the  $c_r$ -balls around  $\min$ ,  $\max$ ,  $u$  and  $v$ . If this is the case, then with very high probability Duplicator can indeed maintain property (4.3), as we will now show. The argument resembles the proof of the classical 0-1-law for first-order logic (cf. [EF99]), but it involves some more housekeeping to deal with the additional structure introduced by the  $+1$ -relation.

Let  $\mu_w$  denote the uniform probability measure on the set  $\mathcal{X}(w, \rho)$ , i.e.,

$$\mu_w(V) := \frac{|V|}{|\mathcal{X}(w, \rho)|}$$

for  $V \subseteq \mathcal{X}(w, \rho)$ . For ease of notation we drop the subscript  $w$ . Let  $s$  be the number of non-isomorphic  $(\sigma \cup \rho)$ -expansions of  $v_{2c_r+2}|_{S^{c_r}(\min, \max, u, v)}$ , and let  $A_1, \dots, A_s$  be structures representing these isomorphism types. Notice that the four  $c_r$ -balls which make up the universe of this substructure do not touch, as is the case in all  $v_n$  and  $w_n$

#### 4 Derandomising Logics

for large enough  $n$ . We let  $V_n^{(j)}$  be the set of all  $(\sigma \cup \rho)$ -expansions  $X$  of  $v_n$  with

$$X|_{S^{c_r}(\min, \max, u, v)} \cong A_j,$$

and analogously for  $W_n^{(j)}$ . If the  $c_r$ -balls around  $\min$ ,  $\max$ ,  $u$  and  $v$  do not touch, then the induced substructures of  $v_n$  and  $w_n$  on the union of these balls are isomorphic. Thus for large enough  $n$ , the  $V_n^{(j)}$  ( $W_n^{(j)}$ ) form a partition of  $\mathcal{X}(v_n, \rho)$  ( $\mathcal{X}(w_n, \rho)$ ), and

$$\mu(V_n^{(j)}) = \mu(W_n^{(j)}) = \frac{1}{s}.$$

For any two structures  $X \in V_n^{(j)}$  and  $Y \in W_n^{(j)}$ , the tuples  $a_1, \dots, a_4$  and  $b_1, \dots, b_4$  as defined above satisfy property (4.3). We now show that there are subsets  $\hat{V}_n^{(j)} \subset V_n^{(j)}$  and  $\hat{W}_n^{(j)} \subset W_n^{(j)}$  such that Duplicator can maintain property (4.3) for  $r$  moves on structures taken from these subsets.

To be precise, we define Duplicator's strategy if Spoiler chooses  $a$  from structure  $X$  as follows:

- (I) If  $a$  is in  $X|_{S^{2c_{r'}+1}(a_1, \dots, a_{k+4})}$ , then choose the corresponding element in  $Y$ , i.e., the unique element  $b \in V(Y)$  which has

$$d_{c_{r'}}(a_i, a) = d_{c_{r'}}(b_i, b)$$

for  $1 \leq i \leq k+4$ . These are exactly the  $a$  whose  $c_{r'}$ -ball touches the  $c_{r'}$ -ball around some previously chosen  $a_i$ .

- (II) Otherwise, choose any element of  $Y$  which has distance  $> 2c_{r'} + 1$  from all elements  $b_1, \dots, b_{k+4}$ . Thus the  $c_{r'}$ -ball around the newly chosen element touches no  $c_{r'}$ -ball around a previously chosen element.

Moves of type (I) in the above strategy can always be carried out by Duplicator and maintain property (4.3). Moves of type (II) can only fail if there is a tuple  $b_1, \dots, b_{k+4}$  in  $Y$  and a  $(\sigma \cup \rho)$ -structure  $Z$  containing elements  $a_1, \dots, a_{k+4}$  and  $a$  such that

- $Z \in \mathcal{X}(v_n, \rho)$ ,
- $Z|_{S^{c_{r'}-k}(a_1, \dots, a_{k+4})} \cong Y|_{S^{c_{r'}-k}(b_1, \dots, b_{k+4})}$ ,
- $d(a, a_i) > 2c_{r'} + 1$  for  $1 \leq i \leq k+4$ , and
- $Z|_{S^{c_{r'}}(a_1, \dots, a_{k+4}, a)} \not\cong Y|_{S^{c_{r'}}(b_1, \dots, b_{k+4}, b)}$  for all  $b \in V(Y)$ .

Let  $m := 3n + 2 = |V(Y)|$ . There are  $O(m^r)$  many possible tuples  $b_1, \dots, b_{k+4}$ , and for each such tuple, there are only constantly (depending only on  $\rho$ ) many choices for  $Z$  and  $a_1, \dots, a_{k+4}, a$  with non-isomorphic  $Z|_{S^{c_{r'}}(a_1, \dots, a_{k+4}, a)}$ . But for each of these  $O(m^r)$  possibilities, there is a subset  $M \subset V(Y)$  with

- $|M| = \Omega(n)$ ,

- $d(b, b_i) > 2c_{r'} + 1$ , for each  $b \in M$  and  $1 \leq i \leq k + 4$ , and
- $d(b, b') > 2c_{r'} + 1$  for every  $b, b' \in M$ .

Because the  $c_{r'}$ -balls around the elements of  $M$  do not overlap, each of the elements in  $M$  satisfies

$$Z|_{S^{c_{r'}}(a_1, \dots, a_{k+4}, a)} \cong Y|_{S^{c_{r'}}(b_1, \dots, b_{k+4}, b)}$$

*independently* with some probability  $p > 0$  depending only on  $r'$  and  $\rho$ . The probability that none of the  $b \in M$  satisfies this is therefore  $(1 - p)^{|M|} = e^{-\Omega(n)}$ , and by a union bound, there is a subset  $\hat{W}_n^{(j)} \subset W_n^{(j)}$  with

$$\mu(\hat{W}_n^{(j)}) = (1 - o(1))\mu(W_n^{(j)})$$

and such that on structures  $Y \in \hat{W}_n^{(j)}$ , Duplicator can maintain property (4.3) for  $r$  many moves when challenged to move in  $Y$ . A subset  $\hat{V}_n^{(j)} \subset V_n^{(j)}$  can be defined analogously.

But now we have defined disjoint sets  $\hat{V}_n^{(1)}, \dots, \hat{V}_n^{(s)} \subset \mathcal{X}(v_n, \rho)$  and  $\hat{W}_n^{(1)}, \dots, \hat{W}_n^{(s)} \subset \mathcal{X}(w_n, \rho)$  such that

- (a)  $|\mu(\hat{V}_n^{(j)}) - \mu(\hat{W}_n^{(j)})| \rightarrow 0$  for  $n \rightarrow \infty$  and all  $1 \leq j \leq s$ ,
- (b)  $\mu\left(\bigcup_j \hat{V}_n^{(j)}\right) \rightarrow 1$  for  $n \rightarrow \infty$
- (c) for every  $n$  and  $j$ , if  $X \in \hat{V}_n^{(j)}$  and  $Y \in \hat{W}_n^{(j)}$ , then  $X \cong_r Y$ .

This implies that for every  $\text{FO}[\sigma \cup \rho]$ -sentence  $\varphi$ ,

$$|\mathbb{P}(v_n \models \varphi) - \mathbb{P}(w_n \models \varphi)| \rightarrow 0$$

as  $n \rightarrow \infty$ , and therefore  $Q$  is not definable in  $\text{BPFO}[+1]$ .  $\square$

## Notes

The results in sections 4.1 and 4.3 have been submitted to CSL2011 [Eic11]. After submitting an earlier version of that paper, we learned from one of the anonymous reviews about the striking similarity between our proof and that of Shelah [She96] which we had been completely unaware of. The extension of Theorem 43 to structures with an equivalence class (i.e., Thm. 47) has been suggested by Anuj Dawar. The results of section 4.2 have been obtained together with Martin Grohe and are published in [EG11].



# Bibliography

- [AB09] Arora, Sanjeev; Barak, Boaz: *Computational Complexity*. Cambridge University Press, 2009.
- [ADF95] Abrahamson, Karl A.; Downey, Rodney G.; Fellows, Michael R.: Fixed-parameter tractability and completeness IV: On completeness for  $W[P]$  and PSPACE analogues. In: *Annals of Pure and Applied Logic*, volume 73:pp. 235–276, 1995.
- [AG07] Alon, Noga; Gutner, Shai: Balanced families of perfect hash functions and their applications. In: *ICALP*, pp. 435–446. 2007.
- [AGHP92] Alon, Noga; Goldreich, Oded; Håstad, Johan; Peralta, René: Simple construction of almost  $k$ -wise independent random variables. In: *Random Struct. Algorithms*, volume 3(3):pp. 289–304, 1992.
- [Ajt83] Ajtai, M.:  $\Sigma_1^1$ -formulae on finite structures. In: *Annals of Pure and Applied Logic*, volume 24(1):pp. 1–48, 1983. ISSN 0168-0072. doi:DOI:10.1016/0168-0072(83)90038-6.
- [AKS98] Alon, Noga; Krivelevich, Michael; Sudakov, Benny: Finding a large hidden clique in a random graph. In: *Random Structures & Algorithms*, volume 13(3-4):pp. 457–466, 1998.
- [AKS04] Agrawal, Manindra; Kayal, Neeraj; Saxena, Nitin: PRIMES is in P. In: *Annals of Mathematics*, volume 160(2):pp. 781–793, 2004.
- [Ama10] Amano, Kazuyuki:  $k$ -subgraph isomorphism on  $AC^0$  circuits. In: *Computational Complexity*, volume 19(2):pp. 183–/210, 2010.
- [AR01] Alekhovich, M.; Razborov, A.: Resolution is not automatizable unless  $W[P]$  is tractable. In: *Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science*, pp. 210–219. 2001.
- [AS92] Alon, Noga; Spencer, Joel: *The Probabilistic Method*. John Wiley, 1992. ISBN 0-471-53588-5.
- [AYZ95] Alon, Noga; Yuster, Raphael; Zwick, Uri: Color-coding. In: *J. ACM*, volume 42:pp. 844–856, July 1995. ISSN 0004-5411. doi:http://doi.acm.org/10.1145/210332.210337.

## Bibliography

- [BBM10] Bosse, Hartwig; Byrka, Jaroslaw; Markakis, Evangelos: New algorithms for approximate nash equilibria in bimatrix games. In: *Theoretical Computer Science*, volume 411(1):pp. 164–173, 2010.
- [BCI<sup>+</sup>10] Borgs, Christian; Chayes, Jennifer; Immorlica, Nicole; Kalai, Adam Tauman; Mirrokni, Vahab; Papadimitriou, Christos: The myth of the folk theorem. In: *Games and Economic Behavior*, volume 70(1):pp. 34 – 43, 2010. Special Issue In Honor of Ehud Kalai.
- [Bea94] Beame, Paul: A switching lemma primer. Technical Report UW-CSE-95-07-01, Department of Computer Science and Engineering, University of Washington, 1994.
- [BES80] Babai, L.; Erdős, P.; Selkow, S.M.: Random graph isomorphism. In: *SIAM Journal on Computing*, volume 9(3):pp. 628–635, 1980.
- [BIS90] Barrington, David A. Mix; Immerman, Neil; Straubing, Howard: On uniformity within NC<sup>1</sup>. In: *J. Comput. Syst. Sci.*, volume 41(3):pp. 274–306, 1990.
- [BL06] Behle, Christoph; Lange, Klaus-Jörn: FO[<]-uniformity. In: *IEEE Conference on Computational Complexity*, pp. 183–189. 2006.
- [BMRV00] Buhrman, H.; Miltersen, P. B.; Radhakrishnan, J.; Venkatesh, S.: Are bitvectors optimal? In: *STOC '00: Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pp. 449–458. ACM, New York, NY, USA, 2000. ISBN 1-58113-184-4. doi:<http://doi.acm.org/10.1145/335305.335357>.
- [Bol01] Bollobás, B.: *Random Graphs*. Cambridge University Press, 2001.
- [Bop97] Boppana, Ravi B.: The average sensitivity of bounded-depth circuits. In: *Information Processing Letters*, volume 63(5):pp. 257–261, 1997.
- [BS95] Boppana, Ravi; Spencer, Joel: Smoothness laws for random ordered graphs. In: Boppana, Ravi; Lynch, James, editors, *Logic and Random Structures*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, pp. 15–32. American Mathematical Society, 1995.
- [CDT09] Chen, Xi; Deng, Xiaotie; Teng, Shang-Hua: Settling the complexity of computing two-player Nash equilibria. In: *J. ACM*, volume 56:pp. 14:1–14:57, 2009.
- [CFI92] Cai, J.-Y.; Fürer, M.; Immerman, N.: An optimal lower bound on the number of variables for graph identifications. In: *Combinatorica*, volume 12(4):pp. 389–410, 1992.

- [CGG06] Chen, Yijia; Grohe, Martin; Grüber, Magdalena: On parameterized approximability. In: *Proceedings of the 2nd International Workshop on Parameterized and Exact Computation*, volume 4169 of *LNCS*, pp. 109–120. Springer-Verlag, 2006.
- [CLRS01] Cormen, Thomas; Leiserson, Charles; Rivest, Ronald; Stein, Clifford: *Introduction to Algorithms*. MIT Press, 2nd edition, 2001.
- [CRVW02] Capalbo, Michael; Reingold, Omer; Vadhan, Salil; Wigderson, Avi: Randomness conductors and constant-degree lossless expanders. In: *CCC '02: Proceedings of the 17th IEEE Annual Conference on Computational Complexity*, p. 15. IEEE Computer Society, Washington, DC, USA, 2002.
- [CS03] Conitzer, Vincent; Sandholm, Tuomas: Complexity results about nash equilibria. In: *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 765–771. Morgan Kaufmann, 2003.
- [DF99] Downey, Rodney G.; Fellows, Michael R.: *Parameterized Complexity*. Springer-Verlag, New York, 1999.
- [DGH<sup>+</sup>02] Dantsin, Evgeny; Goerdt, Andreas; Hirsch, Edward A.; Kannan, Ravi; Kleinberg, Jon; Papadimitriou, Christos; Raghavan, Prabhakar; Schöning, Uwe: A deterministic  $(2 - 2/(k+1))^n$  algorithm for k-sat based on local search. In: *Theoretical Computer Science*, volume 289(1):pp. 69–83, 2002. ISSN 0304-3975. doi:DOI:10.1016/S0304-3975(01)00174-8.
- [DGP09] Daskalakis, Constantinos; Goldberg, Paul W.; Papadimitriou, Christos H.: The complexity of computing a Nash equilibrium. In: *SIAM Journal on Computing*, volume 39(1):pp. 195–259, 2009.
- [DHK95] Dawar, Anuj; Hella, Lauri; Kolaitis, Phokion G.: Implicit definability and infinitary logic in finite model theory. In: *ICALP*, volume 944 of *LNCS*, pp. 624–635. Springer Verlag, 1995.
- [DMP06] Daskalakis, Constantinos; Mehta, Aranyak; Papadimitriou, Christos: A note on approximate nash equilibria. In: *Internet and Network Economics*, volume 4286 of *Lecture Notes in Computer Science*, pp. 297–306. Springer Berlin / Heidelberg, 2006.
- [DMP07] Daskalakis, Constantinos; Mehta, Aranyak; Papadimitriou, Christos H.: Progress in approximate nash equilibria. In: *Proceedings 8th ACM Conference on Electronic Commerce (EC-2007)*, pp. 355–358. ACM, 2007.
- [Ebb85] Ebbinghaus, H.-D.: Extended logics: The general framework. In: Barwise, J.; Feferman, S., editors, *Model-Theoretic Logics*, pp. 25–76. Springer-Verlag, 1985.

- [EF99] Ebbinghaus, H.-D.; Flum, J.: *Finite Model Theory*. Perspectives in Mathematical Logic. Springer-Verlag, 2nd edition, 1999.
- [EFT96] Ebbinghaus, H.-D.; Flum, J.; Thomas, W.: *Einführung in die mathematische Logik*. Spektrum Akademischer Verlag, 4th edition, 1996.
- [EG10] Eickmeyer, Kord; Grohe, Martin: Randomisation and derandomisation in descriptive complexity theory. In: *Computer Science Logic*, volume 6247 of *LNCS*, pp. 275–289. Springer-Verlag, 2010.
- [EG11] Eickmeyer, Kord; Grohe, Martin: Randomisation and derandomisation in descriptive complexity theory. In: *Logical Methods in Computer Science*, 2011.
- [EGG08] Eickmeyer, Kord; Grohe, Martin; Grüber, Magdalena: Approximation of natural w[p]-complete minimisation problems is hard. In: *Proceedings of the 2008 IEEE 23rd Annual Conference on Computational Complexity, CCC '08*, pp. 8–18. IEEE Computer Society, Washington, DC, USA, 2008. ISBN 978-0-7695-3169-4. doi:<http://dx.doi.org/10.1109/CCC.2008.24>.
- [EHV11] Eickmeyer, Kord; Hansen, Kristoffer Arnsfelt; Verbin, Elad: Approximating the minmax value of 3-player games within a constant is as hard as detecting planted cliques, 2011. Submitted to APPROX 2011.
- [Eic11] Eickmeyer, Kord: Non-definability results for randomised first-order logic, 2011. Submitted to CSL 2011.
- [Erd59] Erdős, P.: Graph theory and probability. In: *Canad. J. Math.*, volume 11:pp. 34–38, 1959. ISSN 0008-414X.
- [Fag74] Fagin, Ronald: Generalized first-order spectra and polynomial-time recognizable sets. In: Karp, Richard, editor, *Complexity of Computation*, SIAM-AMS Proceedings, pp. 43–73. 1974.
- [Fag76] Fagin, Ronald: Probabilities on finite models. In: *Journal of Symbolic Logic*, volume 41:pp. 50–58, 1976.
- [Fel57] Feller, W.: *An Introduction to Probability Theory and Its Applications*, volume I. John Wiley & Sons, 1957.
- [FG06] Flum, Jörg; Grohe, Martin: *Parameterized Complexity Theory*. Springer-Verlag, Berlin Heidelberg, 2006.
- [FK00] Feige, Uriel; Krauthgamer, Robert: Finding and certifying a large hidden clique in a semirandom graph. In: *Random Structures & Algorithms*, volume 16(2):pp. 195–208, 2000.
- [FSS81] Furst, Merrick L.; Saxe, James B.; Sipser, Michael: Parity, circuits, and the polynomial-time hierarchy. In: *FOCS*, pp. 260–70. 1981.



- [FSS84] Furst, Merrick L.; Saxe, James B.; Sipser, Michael: Parity, circuits, and the polynomial-time hierarchy. In: *Mathematical Systems Theory*, volume 17(1):pp. 13–7, 1984.
- [GG81] Gabber, Ofer; Galil, Zvi: Explicit constructions of linear-sized superconcentrators. In: *J. Comput. Syst. Sci.*, volume 22(3):pp. 407–420, 1981.
- [GKLT69] Glebskiĭ, Y.V.; Kogan, D.I.; Liogon'kiĭ, M.I.; Talanov, V.A.: Range and degree of realizability of formulas in the restricted predicate calculus. In: *Kibernetika*, volume 2:pp. 17–28, 1969. English translation, *Cybernetics* 5:142–154, 1969.
- [Gro08] Grohe, Martin: The quest for a logic capturing ptime. In: *LICS*, pp. 267–271. 2008.
- [GUV07] Guruswami, Venkatesan; Umans, Christopher; Vadhan, Salil: Unbalanced expanders and randomness extractors from parvaresh-vedya codes. In: *ccc*, volume 00:pp. 96–108, 2007. ISSN 1093-0159. doi:<http://doi.ieeecomputersociety.org/10.1109/CCC.2007.38>.
- [GZ89] Gilboa, Itzhak; Zemel, Eitan: Nash and correlated equilibria: Some complexity considerations. In: *Games and Economic Behavior*, volume 1(1):pp. 80–93, 1989.
- [HHMS08] Hansen, Kristoffer Arnsfelt; Hansen, Thomas Dueholm; Miltersen, Peter Bro; Sørensen, Troels Bjerre: Approximability and parameterized complexity of minmax values. In: *Proceedings of the 4th International Workshop on Internet and Network Economics, WINE 2008*, volume 5385 of *Lecture Notes in Computer Science*, pp. 684–695. Springer, 2008.
- [HK11] Hazan, Elad; Krauthgamer, Robert: How hard is it to approximate the best nash equilibrium? In: *SIAM Journal on Computing*, volume 40(1):pp. 79–91, 2011.
- [HKL96] Hella, L.; Kolaitis, P.G.; Luosto, K.: Almost everywhere equivalence of logics in finite model theory. In: *The Bulletin of Symbolic Logic*, volume 2(4):pp. 422–443, December 1996.
- [Hå86] Håstad, Johan Torkel: *Computational Limitations for Small-Depth Circuits*. Ph.D. thesis, MIT, 1986.
- [Imm82] Immerman, Neil: Upper and lower bounds for first order expressibility. In: *J. Comput. Syst. Sci.*, volume 25(1):pp. 76–98, 1982.
- [Imm99] Immerman, N.: *Descriptive Complexity Theory*. Graduate Texts in Computer Science. Springer-Verlag, 1999.
- [IW97] Impagliazzo, Russell; Wigderson, Avi:  $P = BPP$  if  $E$  requires exponential circuits: Derandomizing the xor lemma. In: *STOC*, pp. 220–229. 1997.

## Bibliography

- [Jer92] Jerrum, Mark: Large cliques elude the metropolis process. In: *Random Structures & Algorithms*, volume 3(4):pp. 347–359, 1992.
- [JP00] Juels, Ari; Peinado, Marcus: Hiding cliques for cryptographic security. In: *Designs, Codes and Cryptography*, volume 20:pp. 269–280, 2000.
- [KL80] Karp, Richard M.; Lipton, Richard J.: Some connections between nonuniform and uniform complexity classes. In: *Proceedings of the twelfth annual ACM symposium on Theory of computing*, STOC '80, pp. 302–309. ACM, New York, NY, USA, 1980. ISBN 0-89791-017-6.
- [Knu81] Knuth, Donald E.: *The Art of Computer Programming*, volume II. Addison-Wesley, 2nd edition, 1981.
- [KPS09] Kontogiannis, Spyros C.; Panagopoulou, Panagiota N.; Spirakis, Paul G.: Polynomial algorithms for approximating nash equilibria of bimatrix games. In: *Theor. Comput. Sci.*, volume 410:pp. 1599–1606, 2009.
- [KS04] Knessl, Charles; Szpankowski, Wojciech: On the number of full levels in tries. In: *Random Struct. Algorithms*, volume 25:pp. 247–276, October 2004. ISSN 1042-9832. doi:10.1002/rsa.20023.
- [Kuč95] Kučera, Luděk: Expected complexity of graph partitioning problems. In: *Discrete Appl. Math.*, volume 57:pp. 193–212, 1995.
- [KV92] Kolaitis, Phokion G.; Vardi, Moshe Y.: Infinitary logics and 0-1 laws. In: *Inf. Comput.*, volume 98(2):pp. 258–294, 1992.
- [KvM02] Klivans, Adam; van Melkebeek, Dieter: Graph nonisomorphism has subexponential size proofs unless the polynomial-time hierarchy collapses. In: *SIAM J. Comput.*, volume 31(5):pp. 1501–1526, 2002.
- [Lau83] Lautemann, C.: BPP and the polynomial hierarchy. In: *Information Processing Letters*, volume 17(4):pp. 215–217, 1983.
- [Lau10] Laubner, Bastian: Capturing polynomial time on interval graphs. In: *LICS*, pp. 199–208. 2010.
- [Lib04] Libkin, L.: *Elements of Finite Model Theory*. Texts in Theoretical Computer Science. Springer-Verlag, 2004.
- [LMM03] Lipton, Richard J.; Markakis, Evangelos; Mehta, Aranyak: Playing large games using simple strategies. In: *ACM Conference on Electronic Commerce*, pp. 36–41. ACM, 2003.
- [LMN89] Linial, N.; Mansour, Y.; Nisan, N.: Constant depth circuits, fourier transform, and learnability. In: *Foundations of Computer Science, Annual IEEE Symposium on*, volume 0:pp. 574–579, 1989. doi:http://doi.ieeecomputersociety.org/10.1109/SFCS.1989.63537.

- [LY94] Lipton, Richard J.; Young, Neal E.: Simple strategies for large zero-sum games with applications to complexity theory. In: *Proceedings of the 26th Annual ACM Symposium on Theory of Computing, STOC'94*, pp. 734–740. ACM Press, 1994.
- [Lyn82] Lynch, J.F.: On sets of relations definable by addition. In: *Journal of Symbolic Logic*, volume 47(3):pp. 659–668, 1982.
- [Mar10] Marx, Dániel: Completely inapproximable monotone and antimonotone parameterized problems. In: *IEEE Conference on Computational Complexity*, pp. 181–187. 2010.
- [MR95] Motwani, Rajeev; Raghavan, Prabhakar: *Randomized Algorithms*. Cambridge University Press, 1995.
- [MS10] Moser, Robin A.; Scheder, Dominik: A full derandomization of schoening’s k-sat algorithm. In: *CoRR*, volume abs/1008.4067, 2010.
- [Mül08] Müller, M.: Valiant-vazirani lemmata for various logics. In: *Electronic Colloquium on Computational Complexity (ECCC)*, volume 15(063), 2008.
- [MV09] Minder, Lorenz; Vilenchik, Dan: Small clique detection and approximate nash equilibria. In: *APPROX and RANDOM 2009: Proceedings of the 12th International Workshop and 13th International Workshop on Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, volume 5687 of *Lecture Notes in Computer Science*, pp. 673–685. Springer, 2009.
- [Nis91] Nisan, Noam: Pseudorandom bits for constant depth circuits. In: *Combinatorica*, volume 11(1):pp. 63–70, 1991.
- [NN93] Naor, Joseph; Naor, Moni: Small-bias probability spaces: Efficient constructions and applications. In: *SIAM J. Comput.*, volume 22(4):pp. 838–856, 1993.
- [NW88] Nisan, Noam; Wigderson, Avi: Hardness vs. randomness. In: *29th Annual Symposium on Foundations of Computer Science*, pp. 2–11. 1988.
- [OR94] Osborne, Martin J.; Rubinstein, Ariel: *A Course in Game Theory*. The MIT Press, 1994.
- [Ott96] Otto, M.: *Bounded Variable Logics and Counting*. Lecture Notes in Logic. Springer-Verlag, 1996.
- [Pap93] Papadimitriou, Christos H.: *Computational Complexity*. Addison Wesley, 1993.
- [Res62] Rescher, N.: Plurality quantification. In: *Journal of Symbolic Logic*, volume 27(3):pp. 373–374, 1962.

## Bibliography

- [Ros08] Rossman, Benjamin: On the constant-depth complexity of  $k$ -clique. In: *Proceedings of the 40th annual ACM symposium on Theory of computing*, STOC, pp. 721–730. 2008. ISBN 978-1-60558-047-0.
- [Ros09] Rossman, Benjamin: Ehrenfeucht-fraïssé games on random structures. In: *WoLLIC*, pp. 350–364. 2009.
- [RVW00] Reingold, Omer; Vadhan, Salil P.; Wigderson, Avi: Entropy waves, the zig-zag graph product, and new constant-degree expanders and extractors. In: *FoCS*, pp. 3–13. 2000.
- [Sax09] Saxena, Nitin: Progress on polynomial identity testing. In: *Electronic Colloquium on Computational Complexity (ECCC)*, volume 16, 2009.
- [Sch02] Schöning, Uwe: A probabilistic algorithm for  $k$ -sat based on limited local search and restart. In: *Algorithmica*, volume 32(4):pp. 615–623, 2002.
- [Sch05] Schweikardt, Nicole: Arithmetic, first-order logic, and counting quantifiers. In: *ACM Trans. Comput. Log.*, volume 6(3):pp. 634–671, 2005.
- [Sch06] Schweikardt, Nicole: On the expressive power of monadic least fixed point logic. In: *Theor. Comput. Sci.*, volume 350(2-3):pp. 325–344, 2006.
- [She96] Shelah, Saharon: Very weak zero one law for random graphs with order and random binary functions. In: *Random Structures & Algorithms*, volume 9(4):pp. 351–358, 1996. ISSN 1098-2418.
- [Sip83] Sipser, Michael: A complexity theoretic approach to randomness. In: *STOC*, pp. 330–335. 1983.
- [Spe01] Spencer, Joel: *The Strange Logic of Random Graphs*. Springer, 2001.
- [Str94] Straubing, Howard: *Finite Automata, Formal Logic, and Circuit Complexity*. Birkhäuser, 1994.
- [TS08] Tsaknakis, Haralampos; Spirakis, Paul G.: An optimization approach for approximate nash equilibria. In: *Internet Mathematics*, volume 5(4):pp. 365–382, 2008.
- [Var82] Vardi, Moshe Y.: The complexity of relational query languages (extended abstract). In: *STOC*, pp. 137–146. 1982.
- [Vio04] Viola, Emanuele: The complexity of constructing pseudorandom generators from hard functions. In: *Computational Complexity*, volume 13:pp. 147–188, 2004.
- [Vio11] Viola, Emanuele: Randomness buys depth for approximate counting, 2011. Available at <http://www.ccs.neu.edu/home/viola/papers/rbd.pdf>.

- [Weg05] Wegener, Ingo: *Complexity Theory*. Springer Verlag, Berlin, 2005.
- [Yao85] Yao, Andrew Chi-Chih: Separating the polynomial-time hierarchy by oracles (preliminary version). In: *FOCS*, pp. 1–10. 1985.



# List of Figures

1.1	Unconditionally known results on BPP. . . . .	14
2.1	Bipartite lossless expanders. . . . .	28
2.2	The overall structure of $\pi(C, k, \delta)$ . . . . .	30
2.3	Layer $\ell$ of $\pi(C, k, \delta)$ . . . . .	31
2.4	Refined gap amplification. . . . .	35
3.1	The gadgets for CFI-graphs . . . . .	58
3.2	The CFI-graph construction for a part of a graph . . . . .	59
3.3	Structures in the class $\mathcal{B}$ . . . . .	61
3.4	The Birthday Paradox . . . . .	62
3.5	The random relation $R$ interpreted as a function. . . . .	64
4.1	A polynomial-size, bounded-depth circuit for $\psi$ . . . . .	70
4.2	Nisan's pseudo-random bit generator . . . . .	76





# Erklärung

Hiermit erkläre ich,

- dass ich die vorliegende Arbeit mit dem Titel „Randomness in Complexity Theory and Logics“ selbständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt und sie an keiner anderen Universität eingereicht habe,
- dass mir die Promotionsordnung der Mathematisch-Naturwissenschaftlichen Fakultät II der Humboldt-Universität Berlin vom 17.01.2005, zuletzt geändert am 13.02.2006, veröffentlicht im Amtlichen Mitteilungsblatt Nr. 34/2006, bekannt ist,
- dass ich keinen Doktorgrad im Fach Informatik besitze.

Berlin, den 20.05.2011

Kord Eickmeyer